KU LEUVEN

ARENBERG DOCTORAL SCHOOL
FACULTY OF ENGINEERING SCIENCE

THE NUMERICAL SOLUTION OF ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS WITH FUZZY COEFFICIENTS

Samuel Corveleyn

Dissertation presented in partial fulfilment of the requirements for the degree of Doctor in Engineering

June 2014

The numerical solution of elliptic partial differential equations with fuzzy coefficients

Samuel CORVELEYN

Examination Committee:
Prof. dr. ir. Pierre Verbaeten, chair
Prof. dr. ir. Stefan Vandewalle, supervisor
Prof. dr. ir. David Moens, supervisor
Prof. dr. ir. Karl Meerbergen
Prof. dr. ir. Giovanni Samaey
Prof. dr. ir. Geert Lombaert
Prof. dr. ir. Gert de Cooman
(Universiteit Gent)
Dr. ir. Jeroen Witteveen
(CWI, Nederland)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Engineering

June 2014

Uitgegeven in eigen beheer, Samuel Corveleyn, Celestijnenlaan 200A box 2402,
B-3001 Heverlee (Belgium)

ISBN 978-94-6018-848-0
D/2014/7515/72

Preface

After my graduation as an Electrotechnical Engineer, I wandered around for some time. I could not decide on whether I wanted to do research in academia, make music, or start working as an engineer. Right before I started this PhD, I came to a point where I had to make a choice. Either I was going to start working as an audio engineer at Philips, or I would start pursuing a PhD under the supervision of Stefan Vandewalle and David Moens. Stefan told me the essential words that a PhD is a once-in-a-lifetime experience. This phrase kept running through my head, and, finally, I made the decision: I am doing it. Oh!, how I sometimes wished I had chosen differently. At times, it was very tough. In the end, I can say, however, that it has been extremely rewarding.

So, first and foremost, I would like to thank Stefan for giving me this opportunity. During my PhD, he was always there when I needed help and advice. I especially experienced a lot of difficulties on transferring my ideas to paper. His careful and patient readings of my drafts were invaluable to me for developing a better, clearer, and more precise writing style. Further, I would like to thank David Moens for getting me interested in fuzzy sets by his enthusiasm and being one of the driving forces behind the *Fuzzy Finite Element Method* IWT SBO project.

I would like to thank Karl Meerbergen, Giovanni Samaey, Geert Lombaert, Gert de Cooman, and Jeroen Witteveen for agreeing to be a member of my PhD examination committee. Their insightful comments and suggestions that followed from a careful reading of my text helped to make it better. Further, I would like to thank Pierre Verbaeten for being the president of the examination committee and Dirk Vandepitte for being a member of my PhD supervisory committee and, as such, guard the progress of my PhD.

I gratefully acknowledge the financial support received from the Flemish agency for Innovation by Science and Technology (IWT-Vlaanderen) in the framework of the *Fuzzy Finite Element Method* SBO project, from the Optimization in

Engineering Center (OPTEC) of Excellence at the KU Leuven, and from the Research Council of the KU Leuven in the framework of the *Computational aspects of uncertainty propagation in large and multiscale systems* OT project.

During the first years at the Computer Science department, I had the honor of being "adopted" by the—although being younger in age—senior generation of PhD students Bart, Joris, Eveline, Pieter, and Liesbeth. They made those years memorable. I had the pleasure of sharing the office with Eveline, Daan, Giacomo, Loïc, Roel, and Peter. They provided the right and quiet atmosphere to work. The later years, I got to know Matthias as a co-organizer of the PhDays. His inimitable humor brought me many enjoyable breaks from the hard work in those last years.

Some other people I would like to thank: Pieter and Sandra for being there when I needed them. Mia for her endless patience and immense support. Tom, Bertoud, and Erwin for being my comrades in engineering. Bert G. for his biting humor. Bart Q. for being my best friend. Bertoud, Dave, and Serge for rocking it out with me every week. Wouter D., Ruben, Alexander, Bernard, and Wouter S. for giving me the chance of beating them in badminton every single time. Wouter S., Bert I., Joke, Levi, Bernard, Pieter, An, Dirk, and Ruben for still being good friends despite me having shared a roof with them for quite some time. Bart V. and Hanne for being inexhaustible sources of wisdom and knowledge. Dries for tirelessly harassing me with the question whether I can solve "thirteenth degree equations" by rote. My parents for being my parents. My brothers for being my brothers. Lauren and Rosie for reminding me that there is more to life than work ... every single day at seven o'clock in the morning.

Finally, I would like to thank Emmily for being there always.

Sam Corveleyn

Abstract

Uncertainty quantification is playing an increasingly important role in the mathematical modeling of physical phenomena. The classical approach to the modeling of uncertainties is to use probability theory. In some cases, it can, however, be argued that probability may not be the most appropriate mathematical representation of the uncertainty. One of the modeling alternatives is then provided by fuzzy sets.

The main theme of this thesis is the study of numerical methods for solving fuzzy differential equations with the fuzzy diffusion equation as our main model problem. To that end, we start with an overview of some of the interpretations and definitions of fuzzy differential equations that can be found in the literature. We conclude from this discussion that the definition using sample path-based fuzzy fields is the most natural one when adhering to the possibilistic interpretation of fuzzy sets. The definition of the solution to a fuzzy differential equation is then defined as the fuzzification by Zadeh's extension principle of the solution operator of a corresponding parametric differential equation.

The practical computation of quantities of interest defined by a fuzzy differential equation amounts to solving a sequence of differential equation constrained global optimization problems over nested search spaces. The large amount of information shared by these optimization problems leaves room for many algorithmic improvements over the independent treatment of these problems.

A first strategy we discuss is the response surface approach. Prior to the computation of the quantities of interest, a response surface approximation of the solution to the parametric differential equation is constructed. An important result we obtain here is that a response surface which is accurate in the L^∞-norm over the parameter domain ensures the accuracy of certain fuzzy quantities of interest derived from it.

We proceed with a detailed discussion of the Galerkin approach to construct a response surface for the parametric diffusion equation. The equation is discretized using finite elements in the spatial domain and a spectral basis of Chebyshev polynomials in the parameter domain. This discretization results in a very large linear algebraic system of Kronecker product structure. Based on ideas from the literature on stochastic diffusion equations, we propose a center-based preconditioner and a multigrid preconditioner. By means of a local Fourier analysis, we show that the convergence properties of the two preconditioners are optimal w.r.t. the discretization parameters. This is confirmed in numerical experiments, which also demonstrate an exponential convergence in the L^∞-norm of the Galerkin approximation as a function of the polynomial degree.

Next, we make abstraction of the underlying problem of solving a fuzzy differential equation and treat the general problem of computing a function of fuzzy numbers. If the function is real-valued and continuous, and if the fuzzy numbers are independent, then the problem amounts to solving a sequence of global minimization and global maximization problems over nested hyperrectangles. Without any extra assumptions on the function, these optimization problems are known to scale exponentially in complexity with the dimension. We make the assumption that the function has some sort of low-rank structure. For such functions, we propose a derivative-free fuzzification algorithm based on a low-rank tensor approximation of the function on a grid followed by the search for the minimal and maximal entries in the low-rank tensor. Numerical experiments on challenging high-dimensional problems show the potential of this method in comparison to some state-of-the-art global optimization routines.

The final part of this thesis is devoted to recycling strategies for the solution of sequences of similar linear systems. Such sequences of systems arise, for example, in the collocation approach to construct a response surface for the parametric diffusion equation. The idea of recycling is that intermediate results or computations of previously solved systems can be used to reduce the computational effort of solving the next system. We propose a method in which the prolongation and restriction operators of the multigrid preconditioner of a reference system are recycled to the other systems. In an algebraic multigrid procedure, the construction of these intergrid transfer operators can amount to a significant portion of the total computation time. A second recycling strategy that we adopt is the reuse of a previous solution as initial guess for the next system. In that regard, it is important to solve the systems in a certain order such that earlier solutions can provide a good initial guess to subsequent ones.

Samenvatting

Onzekerheidskwantificering speelt een steeds belangrijkere rol bij het wiskundig modelleren van fysische verschijnselen. Standaard worden de onzekerheden die voorkomen in deze modellen wiskundig voorgesteld aan de hand van probabilistische modellen. In sommige gevallen kan echter worden gesteld dat een probabilistisch model niet de meest geschikte wiskundige voorstelling is van de onzekerheid in kwestie. Fuzzy (of ook: vage) verzamelingen en fuzzy getallen bieden dan een mogelijk alternatief.

In dit proefschrift ontwikkelen en analyseren we verscheidene numerieke methoden voor het oplossen van fuzzy differentiaalvergelijkingen. De fuzzy diffusievergelijking vormt hierbij onze voornaamste modelprobleem. Om deze studie goed te kunnen uitvoeren geven we eerst een overzicht van enkele van de interpretaties en definities van fuzzy differentiaalvergelijkingen die voorkomen in de literatuur. Wij concluderen uit deze discussie dat de definitie die gebruikmaakt van fuzzy velden gebaseerd op sample-paden de meest natuurlijke is met het oog op de possibilistische interpretatie van fuzzy verzamelingen. De definitie van de oplossing van een fuzzy differentiaalvergelijking wordt dan gedefinieerd als de fuzzificatie van de oplossingsoperator van een overeenkomstige parametrische differentiaalvergelijking door middel van Zadeh's extensieprincipe.

De berekening van numerieke waarden die afhangen van een fuzzy differentiaalvergelijking leidt tot een reeks van globale optimalisatieproblemen over geneste zoekruimten met de parametrische differentiaalvergelijking als beperking. De grote hoeveelheid informatie die gedeeld wordt door deze optimalisatieproblemen laat ruimte voor allerhande algoritmische verbeteringen.

Een eerste strategie die we bespreken is de responsoppervlak-methode. Voorafgaand aan de berekening van numerieke waarden die afhangen van een fuzzy differentiaalvergelijking wordt een responsoppervlak-benadering van de oplossing van de parametrische differentiaalvergelijking opgesteld. Een

belangrijk resultaat dat we hier bekomen is dat een responsoppervlak dat nauwkeurig is in de L^∞-norm over het parameterdomein de nauwkeurigheid garandeert van de fuzzy numerieke waarden die berekend worden aan de hand van dit responsoppervlak.

We vervolgen met een gedetailleerde bespreking van de Galerkin-methode voor het opstellen van het responsoppervlak voor de diffusievergelijking. Deze vergelijking wordt gediscretiseerd met behulp van eindige elementen in het ruimtelijke domein en een spectrale basis van Chebyshev-veeltermen in het parameterdomein. Dit resulteert in een zeer groot lineair algebraïsch stelsel met een Kronecker-productstructuur. Gebaseerd op ideeën uit de literatuur over stochastische diffusievergelijkingen stellen we een centrum-gebaseerde preconditioner en een multigrid-preconditioner voor. Door middel van een lokale Fourieranalyse en numerieke experimenten wordt aangetoond dat de convergentie-eigenschappen van de twee preconditioners optimaal zijn ten opzichte van de ruimtelijke en parametrische discretisatie.

Vervolgens maken we abstractie van het onderliggende probleem van het oplossen van een fuzzy differentiaalvergelijking en behandelen we het algemene probleem van het berekenen van een functie van fuzzy getallen. Als de functie reëelwaardig en continu is, en als de fuzzy getallen onderling onafhankelijk zijn, dan kan het probleem herleid worden tot het oplossen van een reeks globale optimalisatieproblemen over geneste hyperrechthoeken. We maken hier de veronderstelling dat de functie een soort van lage-rang structuur heeft. Voor dergelijke functies stellen we een fuzzificatie-algoritme voor dat begint met de constructie van een tensorbenadering van lage rang van de functie op een rooster. Vervolgens zoekt het algoritme de minimale en maximale waarden in deze tensor om zo de optimalisatieproblemen op te lossen.

Het laatste deel van dit proefschrift is gewijd aan recyclage-strategieën voor het oplossen van sequenties van gelijkaardige lineaire stelsels. Dergelijke sequenties ontstaan bijvoorbeeld bij gebruik van de collocatie-methode voor het opstellen van het responsoppervlak. De idee achter recyclage-strategieën is dat tussenresultaten of tussenbewerkingen van eerder opgeloste stelsels nuttig kunnen zijn voor het efficiënter oplossen van een volgend stelsel. Wij stellen een strategie voor waarbij de prolongatie- en restrictie-operatoren van de multigrid-preconditioner van een referentiestelsel worden hergebruikt bij het opstellen van de multigrid-preconditioners van volgende stelsels. Een tweede strategie die we bespreken is die waarbij een eerdere oplossing herbruikt wordt als initiële schatting van de oplossing van het volgend op te lossen stelsel. Daartoe is het belangrijk dat de stelsels opgelost worden in een welbepaalde volgorde, zodat eerdere oplossingen een goede schatting geven van de oplossing van volgende stelsels.

Contents

List of symbols

Fuzzy sets

α	α-level of an α-cut
$[\widetilde{a}]_\alpha$	α-cut of fuzzy set $\widetilde{a}$; see Definition 2.1.2
$[\widetilde{a}]_\alpha^-$	lower bound of α-cut $[\widetilde{a}]_\alpha$
$[\widetilde{a}]_\alpha^+$	upper bound of α-cut $[\widetilde{a}]_\alpha$
d	metric
d_{H}	Hausdorff distance; see Definition 2.4.1
d_∞	supremum distance; see Definition 2.4.2
$\mathcal{F}(V)$	set of all fuzzy sets over V
$\ominus$	Hukuhara difference operator
$\mu_{\widetilde{a}}$	membership function of fuzzy set $\widetilde{a}$
$\widetilde{a}$	fuzzy sets are denoted by superimposition of a tilde on the set name

Imprecise probabilities

E, F	events
$\mathcal{M}$	credal set
$[\underline{P}, \overline{P}]$	interval probability
$\underline{P}$	necessity measure, lower probability
$\overline{P}$	possibility measure, upper probability
$\mathcal{R}$	set of events
V, W	generic sample spaces

Functional analysis

$C(\Xi)$	shorthand notation for $C(\Xi;\mathbb{R})$
$C(\Xi;X)$	space of X-valued continuous functions on compact (Hausdorff) space Ξ with X a Banach space
$\|u\|_{C(\Xi;X)}$	$C(\Xi;X)$-norm of u $:= \sup_{\xi\in\Xi}\|u(\xi)\|_X$
$D^{\boldsymbol{i}}$	weak partial derivative $:= \dfrac{\partial^{\|\boldsymbol{i}\|_1}.}{\partial x_1^{i_1}\cdots\partial x_d^{i_d}}$
$\|\Omega\|$	diameter $\displaystyle\int_\Omega \mathrm{d}\boldsymbol{x}$ of $\Omega\subset\mathbb{R}^d$
∇	gradient operator $:= (\partial\cdot/\partial x_1,\ldots,\partial\cdot/\partial x_d)$
$H^{-k}(\Omega)$	the continuous dual space of $H_0^k(\Omega)$
$\|q\|_{H^{-k}(\Omega)}$	$H^{-k}(\Omega)$-norm of q $:= \sup_{u\in H_0^k(\Omega),u\neq 0}\dfrac{\|q(u)\|}{\|u\|_{H^k(\Omega)}}$
$H^k(\Omega)$	Sobolev space $:= \{u\in L^2(\Omega)\colon D^{\boldsymbol{i}}u\in L^2(\Omega)\ \forall\|\boldsymbol{i}\|_1\le k\}$
$H_0^k(\Omega)$	Sobolev space $:= \{u\in H^k(\Omega)\colon u\|_{\partial\Omega}=0\}$
$\|u\|_{H^k(\Omega)}$, $\|u\|_{H_0^k(\Omega)}$	$H^k(\Omega)$-norm, $H_0^k(\Omega)$-norm of u $:= \sum_{\|\boldsymbol{i}\|_1\le k}\|D^{\boldsymbol{i}}u\|_{L^2(\Omega)}$
$L^p(\Omega)$	space of Lebesgue measurable functions over $\Omega\subset\mathbb{R}^d$ with finite $L^p(\Omega)$-norm
$\|u\|_{L^p(\Omega)}$	$L^p(\Omega)$-norm of u $:= \begin{cases}\left(\int_\Omega\|u(\boldsymbol{x})\|^p\mathrm{d}\boldsymbol{x}\right)^{\frac{1}{p}} & 1\le p<\infty\\ \operatorname{ess\,sup}_\Omega u & p=\infty\end{cases}$
$L^p_w(\Omega)$	space of weighted Lebesgue measurable functions over $\Omega\subset\mathbb{R}^d$ with finite $L^p_w(\Omega)$-norm
$\|u\|_{L^p_w(\Omega)}$	weighted $L^p(\Omega)$-norm of u $:= \begin{cases}\left(\int_\Omega\|u(\boldsymbol{x})\|^p w(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\right)^{\frac{1}{p}} & 1\le p<\infty\\ \operatorname{ess\,sup}_\Omega u & p=\infty\end{cases}$
w	generic weight function

Vectors, matrices, tensors

$\boldsymbol{A}$	matrices/tensors are denoted by upper-case letters in boldface
$\boldsymbol{A}^+$, $\boldsymbol{A}^-$	matrix splitting of $\boldsymbol{A}$, i.e., $\boldsymbol{A}^+ + \boldsymbol{A}^- = \boldsymbol{A}$
$\boldsymbol{A}_{\|\boldsymbol{i}}$, $A_{\boldsymbol{i}}$	$\boldsymbol{i}$th entry of $\boldsymbol{A}$
$\boldsymbol{A}_{\|\mathcal{J}}$, $A_{\mathcal{J}}$	selection of the $\mathcal{J}$ entries of $\boldsymbol{A}$
$\boldsymbol{A}_{\|\mathcal{J}_1,\ldots,\mathcal{J}_d}$, $A_{\mathcal{J}_1,\ldots,\mathcal{J}_d}$	shorthand notation for $\boldsymbol{A}_{\|\mathcal{J}_1\times\cdots\times\mathcal{J}_d}$
$\cdot$	inner product operator
d	dimension of tensor
$\boldsymbol{I}_n$	Identity matrix of size $n \times n$
$\|\boldsymbol{A}\|_{\mathrm{F}}$	Frobenius norm of $\boldsymbol{A}$ $:= \sqrt{\sum_{\boldsymbol{i}} A_{\boldsymbol{i}}^2}$
$\|\boldsymbol{A}\|_{\infty}$	maximum norm of $\boldsymbol{A}$ $:= \max_{\boldsymbol{i}} \|A_{\boldsymbol{i}}\|$
$\|\boldsymbol{x}\|_p$	so-called p-norm of $\boldsymbol{x}$ (note, however, that $\|\boldsymbol{x}\|_p$, $p < 1$ is not a norm) $:= \begin{cases} \#\{i\colon x_i \neq 0\} & p = 0 \\ \left(\sum_i \|x_i\|^p\right)^{\frac{1}{p}} & 0 < p < \infty \\ \max_i \|x_i\| & p = \infty \end{cases}$
$\odot$	the entrywise Hadamard product operator
$\rho(\boldsymbol{A})$	spectral radius of matrix $\boldsymbol{A}$
$\sigma(\boldsymbol{A})$	spectrum of matrix $\boldsymbol{A}$
$\boldsymbol{x}$	column vectors are denoted by lower-case letters in boldface
$\boldsymbol{x}_{\|i}$, x_i	ith entry of $\boldsymbol{x}$

Index sets

$\mathcal{F}_d$	set of d-dimensional finitely supported multi-indices $:= \{\boldsymbol{i} \in \mathbb{N}_0^d\colon \|\boldsymbol{i}\|_0 < \infty\}$
i, j, k, n, $\boldsymbol{i}$, $\boldsymbol{j}$, $\boldsymbol{n}$	indices
$\mathcal{I}_n$	set of one-dimensional indices $:= \{1, \ldots, n\}$
$\mathcal{I}_n^0$	set of one-dimensional indices $:= \{0, \ldots, n\}$
$\mathcal{I}_{\boldsymbol{n}}$	Cartesian product of sets of one-dimensional indices $:= \bigtimes_{k=1}^{d} \mathcal{I}_{n_k}$

$\mathcal{I}_{\boldsymbol{n}}^0$	Cartesian product of sets of one-dimensional indices $:= \bigtimes_{k=1}^{d} \mathcal{I}_{n_k}^0$
$\mathcal{J}$	generic index set
$\mathcal{J}_{\boldsymbol{w},p}^{\mathrm{SM}}$	index set used in construction of anisotropic sparse grid $:= \bigcup_{\substack{\boldsymbol{j}:=(2^{i_k})_{k=0}^{d} \\ \boldsymbol{i}\in\mathcal{F}_d\colon \boldsymbol{w}\cdot\boldsymbol{i}\le p}} \mathcal{I}_{\boldsymbol{j}}^0$
$\mathcal{J}_{\boldsymbol{w},p}^{\mathrm{TD}}$	index set used in construction of anisotropic space of polynomials of weighted total degree $:= \{\boldsymbol{j}\in\mathcal{F}_d\colon \boldsymbol{j}\cdot\boldsymbol{w}\le p\}$
$\mathcal{J}_{\boldsymbol{w},p}^{\mathrm{TP}}$	index set used in construction of anisotropic space of tensor product polynomials $:= \{\boldsymbol{j}\in\mathcal{F}_d\colon \max_{k=1,\ldots,d_\Xi} j_k w_k \le p\}$
p	total degree in case of Galerkin with $\mathcal{J}_{\boldsymbol{w},p}^{\mathrm{TD}}$, sparse grid level in case of collocation with $\mathcal{J}_{\boldsymbol{w},p}^{\mathrm{SM}}$
$\boldsymbol{w}$	generic weight

Polynomials

Δ_n	operator used in Smolyak construction of $I_{\mathcal{J}}$ $:= I_n - I_{n-1}$
$\Delta_{\boldsymbol{n}}$	operator used in Smolyak construction of $I_{\mathcal{J}}$ $:= \bigotimes_{k=1}^{d_\Xi} \Delta_{n_k}$
$I_{\mathcal{J}}$	Lagrangian interpolation operator for $S_{\mathcal{J}}$, by Smolyak construction: $I_{\mathcal{J}} \equiv \sum_{\boldsymbol{j}\in\mathcal{J}}, \Delta_{\boldsymbol{j}}$
I_n	Lagrangian interpolation operator for S_n
$I_{\boldsymbol{n}}$	Lagrangian interpolation operator for $S_{\boldsymbol{n}}$
$l_{i,n}(\xi)$	Lagrange polynomials $:= \prod_{j=0,j\neq i}^{n} \frac{\xi - s_{j,n}}{s_{i,n} - s_{j,n}}$
$\mathbb{P}_{\mathcal{J}}$	set of polynomials $:= \mathrm{span}\,\{\prod_{k=1}^{d} \xi_k^{j_k} \colon \boldsymbol{j}\in\mathcal{J}\}$
$\mathbb{P}_n$	set of univariate polynomials $:= \mathrm{span}\,\{\xi^0, \ldots, \xi^n\}$
$\mathbb{P}_{\boldsymbol{n}}$	set of tensor product polynomials $:= \bigotimes_{k=1}^{d} \mathbb{P}_{n_k}$

$\mathbb{P}(\Xi; X)$	vector space of polynomials over Ξ with coefficients in X $:= \bigcup_{\mathcal{J} \subset \mathcal{F}_{d_\Xi}} X \otimes \mathbb{P}_{\mathcal{J}}$
s_i	nested univariate interpolation nodes in $[-1, 1]$
$\boldsymbol{s_i}$	nested multivariate interpolation nodes in $[-1, 1]^d$ $:= (s_{i_1}, \ldots, s_{i_d})$
$s_{i,n}$	univariate interpolation nodes in $[-1, 1]$, which are not necessarily nested
$S_{\mathcal{J}}$	set of multivariate interpolation nodes $:= \bigcup_{\boldsymbol{j} \in \mathcal{J}} S_{\boldsymbol{j}}$ $\equiv \{\boldsymbol{s_j} : \boldsymbol{j} \in \mathcal{J}\}$ if S_n, $n = 0, \ldots$ are nested
S_n	set of univariate interpolation nodes $:= \{s_{i,n}\}_{i=0}^{n} \subset [-1, 1]$
$S_{\boldsymbol{n}}$	Cartesian product of sets of univariate interpolation nodes $:= \bigtimes_{k=1}^{d} \mathcal{S}_{n_k}$
$T_j(\xi)$	normalized Chebyshev polynomials $:= \begin{cases} 1 & j = 0\,, \\ \sqrt{2}\cos(j \arccos(\xi)) & j = 1, 2, \ldots\,. \end{cases}$
$T_{\boldsymbol{j}}(\boldsymbol{\xi})$	normalized multivariate Chebyshev polynomials $:= \prod_{k=1}^{d} T_{j_k}(\xi_k)$

Ordinary differential equation

$\triangle t$	time step
f	right hand side function
t	time variable
u	the unknown

Brusselator

A, B, D, E, X, Y	reagents
a, b, d, e, x, y	concentrations of the reagents
$k_1, \ldots, k_4$	reaction rates
x, y	the unknowns

Spatial domain

d_Ω	dimension of spatial domain Ω
Ω	spatial domain, Lipschitz domain in $\mathbb{R}^{d_\Omega}$
$\boldsymbol{x}$	spatial coordinate, element of Ω

Parameter/Uncertainty domain

d_Ξ	dimension of parameter domain Ξ
$\boldsymbol{\xi}$	parameter, element of Ξ
Ξ	parameter or uncertainty domain $:= [-1,1]^{d_\Xi}$

Diffusion equation

a	diffusion coefficient
$\boldsymbol{A}_0, \ldots, \boldsymbol{A}_{d_\Xi}$	matrices following from Galerkin discretization in spatial domain with $\boldsymbol{A}_{0\|i,j} = \int_\Omega a_0(\boldsymbol{x}) \nabla\phi_i(\boldsymbol{x}) \cdot \nabla\phi_j(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x}$, $\boldsymbol{A}_{k\|i,j} = \int_\Omega \sigma a_k(\boldsymbol{x}) \nabla\phi_i(\boldsymbol{x}) \cdot \nabla\phi_j(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x}$, $i, j = 1, \ldots, n_\phi$, $k = 1, \ldots, d_\Xi$
$a_{\max}$	upper bound to a
$a_{\min}$	lower bound to a
$\boldsymbol{f}$	right hand side vector following from Galerkin discretization in spatial domain with $f_i = \int_\Omega f(\boldsymbol{x})\phi_i(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x}$, $i = 1, \ldots, n_\phi$
f	forcing term
$\boldsymbol{g}$	right hand side vector following from Galerkin discretization in parameter space with $g_j = \int_\Xi \psi_j(\boldsymbol{\xi}) w(\boldsymbol{\xi}) \,\mathrm{d}\boldsymbol{\xi}$, $j = 1, \ldots, n_\psi$
$\boldsymbol{G}_0, \ldots, \boldsymbol{G}_{d_\Xi}$	matrices following from Galerkin discretization in parameter space with $\boldsymbol{G}_0 = \boldsymbol{I}_{n_\psi}$, $\boldsymbol{G}_{k\|i,j} = \int_\Xi \xi_k \psi_i(\boldsymbol{\xi}) \psi_j(\boldsymbol{\xi}) w(\boldsymbol{\xi}) \,\mathrm{d}\boldsymbol{\xi}$, $i, j = 1, \ldots, n_\psi$, $k = 1, \ldots, d_\Xi$
$\phi_1, \ldots, \phi_{n_\phi}$	basis functions for finite element approximation space $X_h \subset H_0^1(\Omega)$
$\mathbb{P}_\mathcal{J}$	approximation space for $L_w^2(\Xi)$
$\psi_1, \ldots, \psi_{n_\psi}$	basis functions for approximation space $\mathbb{P}_\mathcal{J} \subset L_w^2(\Xi)$

S	solution operator
u	the unknown
$\boldsymbol{u}$	vector following from discretization of the unknown
u^{r}	numerical approximation of u
$u^{\mathrm{r}}_{n_\phi}(\cdot, \boldsymbol{\xi})$	semi-discrete approximation of $u(\cdot, \boldsymbol{\xi})$ in X_h
$u^{\mathrm{r}}_{n_\phi, n_\psi}$	fully discrete approximation of u in $X_h \otimes \mathbb{P}_{\mathcal{J}}$
$u^{\mathrm{r}}_{n_\psi}$	semi-discrete approximation of u in $H^1_0(\Omega) \otimes \mathbb{P}_{\mathcal{J}}$
X_h	approximation space for $H^1_0(\Omega)$

Elasticity problem

$\boldsymbol{C}$	fourth-order stiffness tensor
E	Young's modulus
$\boldsymbol{\epsilon}$	strain tensor
(f_{x_1}, f_{x_2})	load vector
ν	Poisson's ratio
$\boldsymbol{\sigma}$	stress tensor
(t_{x_1}, t_{x_2})	boundary traction
(u_{x_1}, u_{x_2})	displacement vector describing the deformation

Fuzzy field model

$a_0, \ldots, a_{d_\Xi}$	spatial variations in fuzzy field model; see (3.29)
$C(\boldsymbol{x}_1, \boldsymbol{x}_2)$	covariance function of stochastic field used in modeling of fuzzy field $:= \exp(-\lVert \boldsymbol{x}_1 - \boldsymbol{x}_2 \rVert_1 / L_c)$
$E_0, \ldots, E_{d_\Xi}$	spatial variations in fuzzy field model of Young's modulus
$\kappa_1, \kappa_2, \ldots$	eigenvalues of $C(\boldsymbol{x}_1, \boldsymbol{x}_2)$
L_c	correlation length of $C(\boldsymbol{x}_1, \boldsymbol{x}_2)$
σ	scaling factor in fuzzy field model; see (3.29)
$v_1, v_2, \ldots$	eigenfunctions of $C(\boldsymbol{x}_1, \boldsymbol{x}_2)$

Multigrid

$\boldsymbol{e}_l$	error on level l
$\boldsymbol{f}_l$	right hand side on level l
γ	number of coarse grid corrections (V-cycle: $\gamma = 1$, W-cycle: $\gamma = 2$)
l	level ($l = l_{\max}$ is finest grid, $l = 0$ is coarsest grid)

$\boldsymbol{L}_l$	system matrix on level l
$l_{\max}$	highest level (finest grid)
ν_1, ν_2	number of pre-smoothing and post-smoothing operations
$\boldsymbol{P}_{l-1,l}$	prolongation operator from level $l-1$ to level l
$\boldsymbol{R}_{l,l-1}$	restriction operator from level l to level $l-1$
$\boldsymbol{S}_l$	smoothing operator on level l
$\boldsymbol{u}_l$	current approximation on level l

Local Fourier analysis

$\boldsymbol{e}(\boldsymbol{\theta}, \boldsymbol{z})$	Fourier grid mode
$\boldsymbol{f}_h$	vector following from discretization of the right hand side on grid with spacing h
h	grid spacing
$\mathcal{H}(\boldsymbol{\theta}, \boldsymbol{z})$	subspace of Fourier space $:= \text{span}[\boldsymbol{e}(\boldsymbol{\theta}_1, \boldsymbol{z}), \boldsymbol{e}(\boldsymbol{\theta}_2, \boldsymbol{z}), \boldsymbol{e}(\boldsymbol{\theta}_3, \boldsymbol{z}), \boldsymbol{e}(\boldsymbol{\theta}_4, \boldsymbol{z})]$
λ_m	mth eigenvalue of $\boldsymbol{M}$
$\boldsymbol{L}_h$	matrix following from discretization of the operator on grid with spacing h
$\widehat{\boldsymbol{L}}_h$	Fourier symbol of $\boldsymbol{L}_h$
$\widehat{\mathcal{L}}_h$	action of $\boldsymbol{L}_h$ on $\mathcal{H}(\boldsymbol{\theta}, \boldsymbol{z})$
$\boldsymbol{M}$	matrix from discretization in parameter space with $M_{i,j} := \int_{\Xi} a(\boldsymbol{x}, \boldsymbol{\xi}) \psi_i(\boldsymbol{\xi}) \psi_j(\boldsymbol{\xi}) w(\boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\xi}$ for some $\boldsymbol{x} \in \Omega_h$
ω	Jacobi damping factor
Ω_h	grid with spacing h $:= \{(ih, jh)\}_{i,j \in \mathbb{Z}}$
$\widehat{\boldsymbol{P}}_{2h,2}$	Fourier symbol of prolongation operator
$\widehat{\boldsymbol{R}}_{h,2h}$	Fourier symbol of restriction operator
ρ	convergence factor
$\boldsymbol{S}_h$	smoothing operator on grid with spacing h
$\widehat{\boldsymbol{S}}_h$	Fourier symbol of $\boldsymbol{S}_h$
$\widehat{\boldsymbol{S}}_h^{\mathrm{GS}}$	Fourier symbol of block Gauss–Seidel smoother
$\widehat{\boldsymbol{S}}_h^{\mathrm{JAC}}$	Fourier symbol of block Jacobi smoother
$\widehat{\mathcal{S}}_h$	action of $\boldsymbol{S}_h$ on $\mathcal{H}(\boldsymbol{\theta}, \boldsymbol{z})$
$\widehat{\boldsymbol{T}}_h$	Fourier symbol of two-grid operator
$\boldsymbol{u}_h$	vector following from discretization of the unknown on grid with spacing h
$\boldsymbol{u}_{i,j}$	vector following from discretization of the unknown at grid point (ih, jh)
$\boldsymbol{z}_m$	mth eigenvector of $\boldsymbol{M}$

$\mathcal{H}$-Tucker tensors

f	father of node t
$\mathcal{H}$-Tucker$((r_t)_{t\in T_d})$	set of all tensors of hierarchical rank at most $(r_t)_{t\in T_d}$
l	a level of T_d (root of T_d is at level $l = 0$, deepest level is at $l = l_{\max}$)
$l_{\max}$	depth of T_d
$\mathcal{L}(T_d)$	set of leaves of T_d
r_{eff}	effective rank, defined in relation to storage cost: $(d\, r_{\text{eff}} \max_k n_k + (d-1)r_{\text{eff}}^3)$
$r_{\max}$	maximal rank over all nodes $:= \max_{t\in T_d} r_t$
r_t	rank of node t $:= \operatorname{rank}(\boldsymbol{A}^{(t)})$
$S(t)$	successors of node t
t	a node of T_d
$\bar{t}$	brother of node t
t'	complement of node t $:= \{1,\dots,d\} \setminus t$
T_d	dimension tree
$\mathcal{T}_\epsilon$	$\mathcal{H}$-Tucker leaves-to-root truncation operator up to error ϵ
$\boldsymbol{U}_t$	mode frame at leaf t
$\boldsymbol{V}_t$	transfer tensor at interior node t

Fuzzification algorithm

$\boldsymbol{A}$	low-rank approximation of $\boldsymbol{B}$
$\boldsymbol{B}$	tensor from sampling f on the tensor grid $S_1 \times \cdots \times S_d$, i.e., $B_{\boldsymbol{i}} := f(s_{1,i_1},\dots,s_{d,i_d})$
C	Lipschitz constant of f
ϵ_{appr}	error of fuzzification algorithm $:= d_\infty(f(\widetilde{\boldsymbol{\xi}}), f_{\text{appr}}(\widetilde{\boldsymbol{\xi}}))$
ϵ_{cross}	desired accuracy of ACA approximation
ϵ_{est}	estimated accuracy of ACA approximation
ϵ_{grid}	error in HT algorithm by grid search $:= d_\infty(f(\widetilde{\boldsymbol{\xi}}), f_{\text{grid}}(\widetilde{\boldsymbol{\xi}}))$
ϵ_{opt}	error in HT algorithm by search of extremal elements in low-rank tensor $:= d_\infty(f_{\text{trunc}}(\widetilde{\boldsymbol{\xi}}), f_{\text{opt}}(\widetilde{\boldsymbol{\xi}}))$
ϵ_{trunc}	error in HT algorithm by low-rank approximation $:= d_\infty(f_{\text{grid}}(\widetilde{\boldsymbol{\xi}}), f_{\text{trunc}}(\widetilde{\boldsymbol{\xi}}))$

$[f_{\mathrm{appr}}(\widetilde{\boldsymbol{\xi}})]_\alpha$ approximation of $[f(\widetilde{\boldsymbol{\xi}})]_\alpha$ by fuzzification algorithm

$[f_{\mathrm{grid}}(\widetilde{\boldsymbol{\xi}})]_\alpha$ approximation of $[f(\widetilde{\boldsymbol{\xi}})]_\alpha$ by grid search algorithm
$:= [\min_{\boldsymbol{i}\in\mathcal{I}_{\boldsymbol{n}},\ (s_{1,i_1},\ldots,s_{d,i_d})\in[\widetilde{\boldsymbol{\xi}}]_\alpha} B_{\boldsymbol{i}}, \max_{\boldsymbol{i}\in\mathcal{I}_{\boldsymbol{n}},\ (s_{1,i_1},\ldots,s_{d,i_d})\in[\widetilde{\boldsymbol{\xi}}]_\alpha} B_{\boldsymbol{i}}]$

$[f_{\mathrm{opt}}(\widetilde{\boldsymbol{\xi}})]_\alpha$ approximation of $[f_{\mathrm{trunc}}(\widetilde{\boldsymbol{\xi}})]_\alpha$ by search of extremal entries in low-rank tensor
$:= [\min_{\mathrm{appr},\ \boldsymbol{i}\in\mathcal{I}_{\boldsymbol{n}},\ (s_{1,i_1},\ldots,s_{d,i_d})\in[\widetilde{\boldsymbol{\xi}}]_\alpha} A_{\boldsymbol{i}}, \max_{\mathrm{appr},\ \boldsymbol{i}\in\mathcal{I}_{\boldsymbol{n}},\ (s_{1,i_1},\ldots,s_{d,i_d})\in[\widetilde{\boldsymbol{\xi}}]_\alpha} A_{\boldsymbol{i}}]$

$[f_{\mathrm{trunc}}(\widetilde{\boldsymbol{\xi}})]_\alpha$ approximation of $[f_{\mathrm{grid}}(\widetilde{\boldsymbol{\xi}})]_\alpha$ by low-rank approximation
$:= [\min_{\boldsymbol{i}\in\mathcal{I}_{\boldsymbol{n}},\ (s_{1,i_1},\ldots,s_{d,i_d})\in[\widetilde{\boldsymbol{\xi}}]_\alpha} A_{\boldsymbol{i}}, \max_{\boldsymbol{i}\in\mathcal{I}_{\boldsymbol{n}},\ (s_{1,i_1},\ldots,s_{d,i_d})\in[\widetilde{\boldsymbol{\xi}}]_\alpha} A_{\boldsymbol{i}}]$

$f(\widetilde{\boldsymbol{\xi}})$ fuzzy number to compute by α-cut approach

h_k spacing between the n_k sample points in the kth dimension
$:= ([\widetilde{\xi}_k]_0^+ - [\widetilde{\xi}_k]_0^-)/(n_k - 1)$

N_{eval} number of function evaluations used by fuzzification algorithm

n_{iter} number of iterations in pivot search in ACA algorithm

$\boldsymbol{p}_1^t, \ldots, \boldsymbol{p}_{r_t}^t$ pivots at node t used in ACA algorithm

P^t set of pivots at node t used in ACA algorithm
$:= \{\boldsymbol{p}_1^t, \ldots, \boldsymbol{p}_{r_t}^t\}$

S_k set of n_k equidistantly distributed sample points in the kth dimension
$:= \{s_{k,i_k} := [\widetilde{\xi}_k]_0^- + (i_k - 1)h_k\}_{i_k=1}^{n_k}$

$s_{k,1}, \ldots, s_{k,n_k}$ equidistantly distributed sample points in the kth dimension

List of acronyms

$\mathcal{H}$-Tucker	hierarchical Tucker
ACA	adaptive cross approximation
AMG	algebraic multigrid
CG	conjugate gradient
DE	differential equation
DIRECT	Dividing Rectangles algorithm
HT	Hierarchical Tucker algorithm
INTERALG	Interval Algorithm
LFA	local Fourier analysis
MCS	Multilevel Coordinate Search algorithm
ODE	ordinary differential equation
PDE	partial differential equation

Chapter 1

Introduction

1.1 Uncertainty quantification

Mathematical simulations of physical phenomena play an increasingly important role in science and engineering. Due to the ever growing computational power, the models used in these simulations are becoming more detailed and more accurate representations of reality. The question rises then, however, whether it makes any sense to perform such detailed deterministic simulations when in practice there are often many ill-known and uncertain parameters. Relying on simulations that ignore the uncertainties can give a false sense of precision. As such, the incorporation of the uncertainties into the model may allow for a better assessment of the accuracy and the quality of the simulation results. In engineering applications, these uncertainties can be used to estimate the reliability of a design, to do a risk analysis, to make a robust design, etcetera. The quantification and mathematical description of uncertainties in physical phenomena is, however, not an easy task. First, there is the philosophical question of what uncertainty is and how to correctly capture it in a mathematical structure. Second, quantifying the uncertainties into the chosen mathematical structure from measurements and expert knowledge can be very hard and time-consuming in practice.

Uncertainties come in different forms and from different sources. A well accepted way of classifying uncertainties is to make a distinction between *aleatoric* and *epistemic* uncertainty [166]. Aleatoric uncertainty is regarded as inherent to the physics of the problem. It is supposed to be irreducible, no matter the amount of effort spent on measurements and experiments.

Epistemic uncertainty on the other hand originates from a lack of knowledge. It can come from insufficient data, a poor understanding of the physics, vagueness in linguistic descriptions of parameters, undecided details occurring in the early stages of an engineering design, etcetera. Such uncertainties are supposed to be reducible. A crisp distinction between aleatoric and epistemic is seldomly possible and uncertainties will often be a mix of both. Moreover, some argue that all uncertainty is of a purely epistemic nature [48].

The classical and most accepted way to mathematically describe uncertainty is to use probability. In the case of aleatoric uncertainty, this corresponds to the frequentist view [202] and the propensity view [172, 173] on uncertainty. The probability density then expresses, respectively, the frequency or chance of occurrence of a certain outcome of an experiment. When dealing with epistemic uncertainty, the probability density expresses a subjective degree of belief in a certain outcome. This is the subjectivist or Bayesian interpretation of probability [40, 47, 112, 176]. Knowledge is cast in a prior probability density and can subsequently be updated using Bayesian inference when new data becomes available.

Information or knowledge about a phenomenon may, however, be incomplete, contradicting, vague, etcetera. In that case, it can be argued that probability may not be the most appropriate mathematical representation of the uncertainty [61, 66, 96, 95, 123, 154, 180, 221]. The reasoning generally used is that the mathematical description of the uncertainty should be able to represent the lack of information and the type of information as given. If, for example, a real parameter is only known to be within certain bounds, then an interval is a perfect representation of this uncertainty. From a subjectivist Bayesian point of view, such an uncertainty would, for example, have been modeled with a uniform prior probability distribution based on Laplace's principle of insufficient reason [105].

Interval uncertainty is an example of a simple uncertainty model with a limited expressive power. Many other and more versatile uncertainty models have been introduced over the last decades. Although very different at first sight, some successful attempts have been made at unifying many of them in a general uncertainty theory [122, 206, 212, 221]. An important and often used unifying theory is the theory of imprecise probabilities [206, 207, 208]. It encompasses interval [155] and fuzzy [218] uncertainties, possibility theory [41, 42, 43, 220], Dempster–Shafer theory [53, 187] and probability theory. These theories lead to a definition of upper and lower probabilities, or more generally, interval probabilities [212]. Moreover, also other uncertainty models which do not necessarily allow a description by interval probabilities, like probability boxes [54] and credal sets [138], fit into the imprecise probability framework.

1.2 Fuzzy sets

In this thesis, we confine ourselves to uncertainties modeled by fuzzy sets. Such sets will be discussed in more detail in Chapter 2. Here, we will mention some of the key concepts and terminology, which is required for a basic understanding of our research goals and contributions. Fuzzy sets were introduced by Zadeh in [218] as a way to represent vague linguistic knowledge in logic. These sets extend the definition of a classical set by allowing a gradual transition in membership degree between 0 (not a member of) and 1 (a member of). As such, a fuzzy set $\widetilde{a}$ over a set V is represented by a membership function $\mu_{\widetilde{a}}\colon V \to [0,1]$.

Central in fuzzy computations is the definition of operations on fuzzy sets by the extension of functions operating on deterministic elements. Given a function $f\colon V \to W$, Zadeh's extension principle [219] defines the membership of $f(\widetilde{a})$ as

$$\mu_{f(\widetilde{a})}(z) = \begin{cases} \sup_{a \in f^{-1}(z)} \mu_{\widetilde{a}}(a) & f^{-1}(z) \neq \emptyset\,, \\ 0 & f^{-1}(z) = \emptyset\,. \end{cases}$$

This extension principle can be hard to apply in numerical computations. By means of the so-called α-cuts defined as $[\widetilde{a}]_\alpha := \{a \in V\colon \mu_{\widetilde{a}}(a) \geq \alpha\}$, it can, however, be reformulated as

$$[f(\widetilde{a})]_\alpha = f([\widetilde{a}]_\alpha)\,, \quad \text{for all } \alpha \in [0,1]\,,$$

if f is continuous, $\mu_{\widetilde{a}}$ is upper semi-continuous, and $[\widetilde{a}]_0$ is compact [159]. Further assuming that f is a real-valued function, i.e., $W = \mathbb{R}$, the extension principle leads to a sequence of global optimization problems

$$[f(\widetilde{a})]_\alpha = [\min_{a \in [\widetilde{a}]_\alpha} f(a), \max_{a \in [\widetilde{a}]_\alpha} f(a)]\,, \quad \text{for all } \alpha \in [0,1]\,. \tag{1.1}$$

1.3 Main research goals, contributions and outline of the thesis

The objective of this thesis is to develop and analyze numerical methods for solving fuzzy differential equations (DE). The main model problem that we consider in this work is the steady-state diffusion equation

$$-\nabla \cdot \left(a(\boldsymbol{x}, \widetilde{\boldsymbol{\xi}}) \nabla u(\boldsymbol{x}, \widetilde{\boldsymbol{\xi}}) \right) = f(\boldsymbol{x}) \tag{1.2}$$

on a d_Ω-dimensional Lipschitz domain $\Omega \subset \mathbb{R}^{d_\Omega}$ with f a deterministic source term, function of $\boldsymbol{x} \in \Omega$. The fuzzy input field a and the unknown fuzzy solution field u depend on the spatial coordinate $\boldsymbol{x}$ and on a fuzzy vector $\widetilde{\boldsymbol{\xi}}$. We assume that the latter has hyperrectangular α-cuts that are subsets of $[-1,1]^{d_\Xi}$. Furthermore, we complete (1.2) with deterministic (typically homogeneous) Dirichlet boundary conditions.

Chapter 2. Preliminaries on fuzzy set theory, fuzzy numbers and arithmetic.

Chapter 2 will provide some background on fuzzy set theory. We go into detail on the definition of fuzzy sets and discuss some key concepts of fuzzy set theory. We end the chapter with a discussion of the interpretation of a fuzzy set as an imprecise probability model. This will allow for a better understanding of fuzzy sets as a model for uncertainty.

Chapter 3. Fuzzy differential equations.

Chapter 3 starts with an overview of some of the possible interpretations and definitions of fuzzy DEs [26, 27, 147, 165, 204]. In this thesis, we will make use of the definition of fuzzy DEs which is based on sample path-based fuzzy fields and Zadeh's extension principle [10, 36, 137, 165]. Using this approach, the solution to the fuzzy partial differential equation (PDE) (1.2) will be defined as the fuzzified solution (i.e., by Zadeh's extension of the solution operator) of the deterministic parametric PDE

$$-\nabla \cdot \big(a(\boldsymbol{x},\boldsymbol{\xi})\nabla u(\boldsymbol{x},\boldsymbol{\xi})\big) = f(\boldsymbol{x})\,. \tag{1.3}$$

The definition using sample path-based fuzzy fields and Zadeh's extension principle matches with the approach used in the fuzzy finite element method [3, 29, 150, 177]. Yet, there are only few papers in the literature that touch on the subject of the definition and interpretation of fuzzy DEs and fuzzy derivatives using sample path-based fuzzy fields. Moreover, when a numerical solution is considered, it is found that a consistent and solid mathematical framework is often missing. In Chapter 3, we attempt to provide such a framework.

We show that a numerical solution to the parametric PDE (1.3) which is accurate in the L^∞-norm over the parameter domain can be used to construct a fuzzy solution to the fuzzy PDE (1.2) which is accurate in some well defined fuzzy sense. This matches with other findings in the literature on the solution of DEs with epistemic uncertainty [67, 78, 103] and is in contrast with the stochastic case [4, 6, 214], where weighted L^2 accuracy w.r.t. the probability

density function is required. We end the chapter with a discussion of some numerical methods to solve the parametric PDE (1.3). More particularly, we go into detail on the collocation and the Galerkin approach.

Chapter 4. Preconditioners for the Galerkin system.

In Chapter 4, we focus on the Galerkin approach to solve the parametric PDE (1.3). The equation is discretized using finite elements in the spatial domain and a spectral basis of Chebyshev polynomials in the parameter domain. This discretization results in a very large linear algebraic system of Kronecker product structure. Based on ideas from the literature on stochastic diffusion equations, we propose two preconditioners for solving this system in combination with a Krylov subspace method. The first preconditioner is the fuzzy analogue to the well-known mean-based preconditioner [174], whereas the second one is of multigrid type closely related to the algorithms in [68, 133, 186]. By means of a local Fourier analysis (LFA), we show that the convergence properties of the two preconditioners are optimal w.r.t. the discretization parameters, i.e., the size of the spatial mesh and the degree of Chebyshev polynomials. We end the chapter with numerical experiments that demonstrate the accuracy of the Galerkin approximation and the efficiency of the proposed iterative solvers.

Chapter 5. Low-rank tensor based methods for fuzzification.

When faced with a fuzzy DE in an engineering context, the problem often boils down to the computation of certain quantities of interest. In the case of model problem (1.2), such quantities of interest could, for example, be the pointwise value $u(\boldsymbol{x}, \widetilde{\boldsymbol{\xi}})$ of the fuzzy field at some point $\boldsymbol{x} \in \Omega$ or the average value $\frac{1}{|\Omega|}\int_\Omega u(\boldsymbol{x}, \widetilde{\boldsymbol{\xi}})\mathrm{d}(\boldsymbol{x})$. Following the interpretation of fuzzy DEs using sample path-based fuzzy fields, the computation of, e.g., $u(\boldsymbol{x}, \widetilde{\boldsymbol{\xi}})$, through the computation of its α-cuts (see (1.1)), leads to a sequence of global optimization problems

$$[u(\boldsymbol{x}, \widetilde{\boldsymbol{\xi}})]_\alpha = [\min_{\boldsymbol{\xi}\in[\widetilde{\boldsymbol{\xi}}]_\alpha} u(\boldsymbol{x}, \boldsymbol{\xi}), \max_{\boldsymbol{\xi}\in[\widetilde{\boldsymbol{\xi}}]_\alpha} u(\boldsymbol{x}, \boldsymbol{\xi})], \quad \text{for all } \alpha \in [0, 1],$$

over nested hyperrectangular search spaces $[\widetilde{\boldsymbol{\xi}}]_\alpha \subset [-1, 1]^{d_\Xi}$.

A big challenge in such global optimization problems is the so-called curse of dimensionality [16, 188]. This means that the computational work of solving the optimization problems scales excessively with the dimension d_Ξ. Confining ourselves to derivative-free optimization and an objective function ($\boldsymbol{\xi} \mapsto u(\boldsymbol{x}, \boldsymbol{\xi})$

in the example above) which is only known to be Lipschitz continuous, it is known that the exponentially scaling grid sampling algorithm (i.e., choosing the optimal function value from function values obtained by sampling the function on an equidistant tensor grid) is optimal [164, 201]. To break the curse of dimensionality, more assumptions on the function than mere Lipschitz continuity are therefore necessary.

The extra assumption on the function that we make in Chapter 5 is that the tensor generated by the tensor grid sampling of the objective function can be well approximated by a low-rank tensor. For such functions, we propose a fuzzification algorithm based on grid search which consists of two stages. In a first stage, our algorithm constructs a low-rank tensor approximation over the α-cut $[\widetilde{\boldsymbol{\xi}}]_0$ using an algorithm from [9]. In a second stage, the actual optimization on different α-levels is done using the method from [71, 127]. This method returns an approximation of the minimal and the maximal entry in a low-rank tensor. As a demonstration of the potential of the proposed method, we compare our algorithm to some state-of-the-art derivative-free global optimization routines in a numerical test with some challenging high-dimensional optimization problems.

Chapter 6. Recycling strategies for the solution of sequences of similar linear systems.

When using the collocation approach to solve the parametric PDE (1.3), a linear system—which results from a spatial discretization of the parametric PDE—needs to be solved for each collocation point considered. These systems and the corresponding solutions are typically very similar. A considerable reduction in computational cost can then be achieved if some information is recycled from one system to the next system. Most of the recycling techniques found in the literature are based on the premise that an iterative solver is used. In that case, reusing the previous solution as initial guess for the next system [73], reusing the preconditioner constructed for some reference system [69, 174, 199], and recycling Krylov subspace vectors [28, 168] are possible strategies.

We propose a method in which the prolongation and restriction operators of the multigrid preconditioner of one reference system are recycled to the other systems. In an algebraic multigrid (AMG) procedure, the construction of these intergrid transfer operators can amount to a significant portion of the total computation time. By recycling them, a preconditioner can be constructed which is nearly as good as a finely tuned AMG preconditioner while being much cheaper to construct. A second recycling strategy that we adopt is the reuse of a previous solution as initial guess for the next system. In that regard,

it is important to solve the systems in such an order that earlier solutions can provide a good initial guess for subsequent ones [81]. We discuss such an ordering strategy. To demonstrate the proposed recycling strategies, we end the chapter with a numerical experiment.

Chapter 7. Conclusions.

In chapter 7, we formulate the conclusions of this work, summarize the main contributions, and indicate possible future research directions.

1.4 Some notation

Tensors and matrices will be denoted by a capital bold letter. Vectors (in the meaning of column matrices) will be denoted by a small bold letter. Let $\mathcal{I}_{\boldsymbol{n}}$ be the set of multi-indices defined as

$$\mathcal{I}_{\boldsymbol{n}} := \bigtimes_{k=1}^{d} \mathcal{I}_{n_k} \quad \text{with} \quad \mathcal{I}_n := \{1, \ldots, n\}, \tag{1.4}$$

and let $\boldsymbol{A} \in \mathbb{R}^{\mathcal{I}_{\boldsymbol{n}}}$. For $\mathcal{J} \subset \mathcal{I}_{\boldsymbol{n}}$, the notation $\boldsymbol{A}_{|\mathcal{J}}$ then represents the selection of the corresponding entries of $\boldsymbol{A}$. When no confusion is possible, we will also use the notation $A_{\mathcal{J}}$. As a shorthand notation for $A_{\mathcal{J}_1 \times \cdots \times \mathcal{J}_d}$ with $\mathcal{J}_k \subset \mathcal{I}_k$, $k = 1, \ldots, d$, we will use $A_{\mathcal{J}_1, \ldots, \mathcal{J}_d}$. In the case that $\mathcal{J}_k$ is a singleton, the element may be written as a shorthand notation for the set, and in the case that $\mathcal{J}_k = \mathcal{I}_k$, a dot '$\cdot$' is used to denote the whole set. This means that $A_{i_1,\ldots,i_d}$ represents the $(i_1, \ldots, i_d)$th entry of $\boldsymbol{A}$ and $A_{i_1,\ldots,i_{k-1},\cdot,i_{k+1},\ldots,i_d} = A_{i_1,\ldots,i_{k-1},\mathcal{I}_k,i_{k+1},\ldots,i_d}$.

The notation $\|\boldsymbol{x}\|_p$ for a vector $\boldsymbol{x} \in \mathbb{R}^{\mathcal{I}_n}$ is used to denote its so-called p-norm. It is defined as

$$\|\boldsymbol{x}\|_p := \begin{cases} \#\{i \in \mathcal{I}_n \colon x_i \neq 0\} & p = 0, \\ \left(\sum_{i \in \mathcal{I}_n} |x_i|^p\right)^{\frac{1}{p}} & 0 < p < \infty, \\ \max_{i \in \mathcal{I}_n} |x_i| & p = \infty. \end{cases} \tag{1.5}$$

Remark, however, that for $0 \leq p < 1$, $\|\cdot\|_p$ is not a true norm (the triangle inequality does not hold). For a matrix/tensor $\boldsymbol{A} \in \mathbb{R}^{\mathcal{I}_{\boldsymbol{n}}}$ we use the notation $\|\boldsymbol{A}\|_\infty$ to denote the maximum norm

$$\|\boldsymbol{A}\|_\infty := \max_{\boldsymbol{i} \in \mathcal{I}_{\boldsymbol{n}}} |A_{\boldsymbol{i}}|, \tag{1.6}$$

and $\|\boldsymbol{A}\|_{\mathrm{F}}$ to denote the Frobenius norm

$$\|\boldsymbol{A}\|_{\mathrm{F}} := \sqrt{\sum_{\boldsymbol{i}\in\mathcal{I}_{\boldsymbol{n}}} A_{\boldsymbol{i}}^2}\,. \tag{1.7}$$

For vectors, we will use the following partial order relation:

$$\boldsymbol{x} \leq \boldsymbol{y} \Leftrightarrow x_i \leq y_i \text{ for all } i \in \mathcal{I}_n. \tag{1.8}$$

Finally, the identity matrix of size $n \times n$ will be denoted by $\boldsymbol{I}_n$.

Further, we will make use of the following Banach spaces:

- $L^p(\Omega)$: space of Lebesgue measurable functions over $\Omega \subset \mathbb{R}^d$ with finite $L^p(\Omega)$-norm.
- $L^p_w(\Omega)$: space of weighted Lebesgue measurable functions over $\Omega \subset \mathbb{R}^d$ with finite $L^p_w(\Omega)$-norm.
- Sobolev space $H^k(\Omega) := \{u \in L^2(\Omega)\colon D^{\boldsymbol{i}}u \in L^2(\Omega)\ \forall \|\boldsymbol{i}\|_1 \leq k\}$.
- Sobolev space $H^k_0(\Omega) := \{u \in H^k(\Omega)\colon u|_{\partial\Omega} = 0\}$.
- $H^{-k}(\Omega)$: the continuous dual space of $H^k_0(\Omega)$.
- $C(\Xi; X)$: space of X-valued continuous functions on compact (Hausdorff) space Ξ with X a Banach space.
- $C(\Xi)$: shorthand notation for $C(\Xi; \mathbb{R})$.

Here, the symbol $D^{\boldsymbol{i}}$ is used to denote the weak partial derivative $\dfrac{\partial^{\|\boldsymbol{i}\|_1}.}{\partial x_1^{i_1}\cdots\partial x_d^{i_d}}$.
The norms used for these Banach spaces are:

- $\|u\|_{L^p(\Omega)} := \begin{cases} \left(\int_\Omega |u(\boldsymbol{x})|^p \mathrm{d}\boldsymbol{x}\right)^{\frac{1}{p}} & 1 \leq p < \infty, \\ \operatorname{ess\,sup}_\Omega u & p = \infty. \end{cases}$
- $\|u\|_{L^p_w(\Omega)} := \begin{cases} \left(\int_\Omega |u(\boldsymbol{x})|^p w(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}\right)^{\frac{1}{p}} & 1 \leq p < \infty, \\ \operatorname{ess\,sup}_\Omega u & p = \infty. \end{cases}$
- $\|u\|_{H^k(\Omega)} := \sum_{\|\boldsymbol{i}\|_1 \leq k} \|D^{\boldsymbol{i}}u\|_{L^2(\Omega)}$, $\|u\|_{H^k_0(\Omega)} := \sum_{\|\boldsymbol{i}\|_1 \leq k} \|D^{\boldsymbol{i}}u\|_{L^2(\Omega)}$.

- $\|q\|_{H^{-k}(\Omega)} := \sup_{u \in H_0^k(\Omega), u \neq 0} \frac{|q(u)|}{\|u\|_{H_0^k(\Omega)}}.$
- $\|u\|_{C(\Xi;X)} := \sup_{\xi \in \Xi} \|u(\xi)\|_X.$

Chapter 2

Preliminaries on fuzzy set theory, fuzzy numbers and arithmetic

This chapter recalls some necessary background material and basic concepts from fuzzy set theory. For a more thorough treatise on fuzzy sets and fuzzy numbers, we refer to [111, 180].

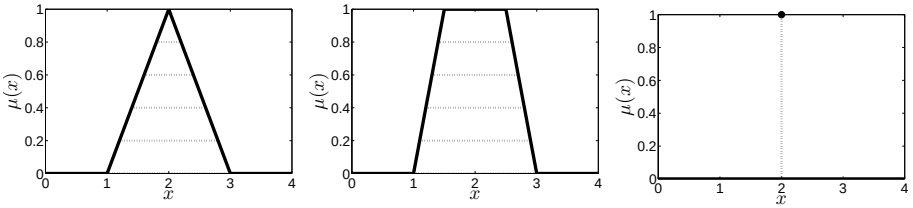
2.1 Fuzzy sets and fuzzy numbers

Fuzzy sets generalize the concept of classical sets. Classical sets either contain an element or not. This is reflected in the fact that the indicator function of a classical set only takes values 0 or 1. Fuzzy sets, on the other hand, allow any degree of membership between 0 and 1. As such, a fuzzy set is represented by a membership function that can take any value in the interval $[0, 1]$.

Definition 2.1.1 (Fuzzy set, membership function)**.** *A fuzzy set $\widetilde{a}$ is a pair $(V, \mu_{\widetilde{a}})$, where V is a set and $\mu_{\widetilde{a}} : V \to [0, 1]$ is a membership function.*

Fuzzy sets will be denoted by the superimposition of a tilde on the set name. The set of all fuzzy sets over a set V will be denoted by $\mathcal{F}(V)$.

An important notion in fuzzy set theory is that of α-cuts.



(a) triangular (b) trapezoidal (c) crisp

Figure 2.1: Examples of fuzzy numbers with illustrations of some of their α-cuts.

Definition 2.1.2 (α-cut)**.** *The α-cuts of a fuzzy set $\widetilde{a} = (V, \mu_{\widetilde{a}})$ over a topological space V are defined as*

$$[\widetilde{a}]_\alpha := \begin{cases} \left\{ a \in V : \mu_{\widetilde{a}}(a) \geq \alpha \right\} & \alpha \in (0,1]\,, \\ \text{closure}\left(\bigcup_{\alpha \in (0,1]} [\widetilde{a}]_\alpha \right) & \alpha = 0\,. \end{cases}$$

The $\alpha = 0$ cut and $\alpha = 1$ cut are, respectively, called the (topological) support and the core of the fuzzy set.

In this thesis, we will mostly work with fuzzy sets that are normal, compact, and defined over $\mathbb{R}$. Such sets will be called fuzzy numbers. In the literature, they are sometimes also referred to as fuzzy intervals.

Definition 2.1.3 (Normal fuzzy set)**.** *A fuzzy set $\widetilde{a}$ is called normal if $[\widetilde{a}]_1$ is nonempty.*

Definition 2.1.4 (Compact fuzzy set)**.** *A fuzzy set $\widetilde{a}$ is called compact if $[\widetilde{a}]_\alpha$ is compact for all $\alpha \geq 0$.*

Definition 2.1.5 (Fuzzy number)**.** *A fuzzy set $\widetilde{a} \in \mathcal{F}(\mathbb{R})$ which is normal and compact is called a fuzzy number.*

The upper and lower bound of the α-cut $[\widetilde{a}]_\alpha$ of a fuzzy number $\widetilde{a}$ will be denoted by $[\widetilde{a}]_\alpha^+$ and $[\widetilde{a}]_\alpha^-$, i.e., we have $[\widetilde{a}]_\alpha \equiv [[\widetilde{a}]_\alpha^-, [\widetilde{a}]_\alpha^+]$.

Example 2.1.1. *Three examples of typical fuzzy numbers are shown in Fig. 2.1: a triangular, trapezoidal, and crisp number respectively, where the latter is actually the fuzzy representation of a classical number.*

2.2 Functions of fuzzy sets and Zadeh's extension principle

The definition of operations on fuzzy numbers, or more generally operations on fuzzy sets, is often based on one of the fundamental axioms in fuzzy set theory, namely *Zadeh's extension principle* [219]. This principle defines how a map $f\colon V \to W$ operating on *deterministic* elements of V should be extended towards a map $f\colon \mathcal{F}(V) \to \mathcal{F}(W)$, operating on fuzzy sets in $\mathcal{F}(V)$. It is the so-called fuzzification of that map.

Definition 2.2.1 (Zadeh's extension principle)**.** *If $f\colon V \to W$, then $f\colon \mathcal{F}(V) \to \mathcal{F}(W)$ is defined by $f\colon \widetilde{a} = (V, \mu_{\widetilde{a}}) \mapsto f(\widetilde{a}) = (W, \mu_{f(\widetilde{a})})$ with*

$$\mu_{f(\widetilde{a})}(z) = \begin{cases} \sup_{a \in f^{-1}(z)} \mu_{\widetilde{a}}(a) & f^{-1}(z) \neq \emptyset\,, \\ 0 & f^{-1}(z) = \emptyset\,. \end{cases} \tag{2.1}$$

The explicit mention of the case $f^{-1}(z) = \emptyset$ is not necessary by definition of sup as smallest upper bound, i.e., $\sup_{a \in \emptyset} \mu_{\widetilde{a}}(a)$ is the bottom of $[0,1]$, so 0. For that reason, we will further on always omit the case $f^{-1}(z) = \emptyset$ when writing down Zadeh's extension principle.

The direct application of the extension principle to compute a function of a fuzzy set numerically can be quite cumbersome. Fortunately, such a computation can be simplified considerably in many cases. It is proved in [159, Proposition 5.1] that the α-cuts $[f(\widetilde{a})]_\alpha$ are equal to $f([\widetilde{a}]_\alpha)$, if f is continuous, $\mu_{\widetilde{a}}$ is upper semi-continuous, and $[\widetilde{a}]_0$ is compact.

Theorem 2.2.1 (α-cut approach)**.** *Let $f\colon V \to W$ be a continuous map and $\widetilde{a} \in \mathcal{F}(V)$ a fuzzy set with compact support and upper semi-continuous membership function. Then, the Zadeh extension of f satisfies*

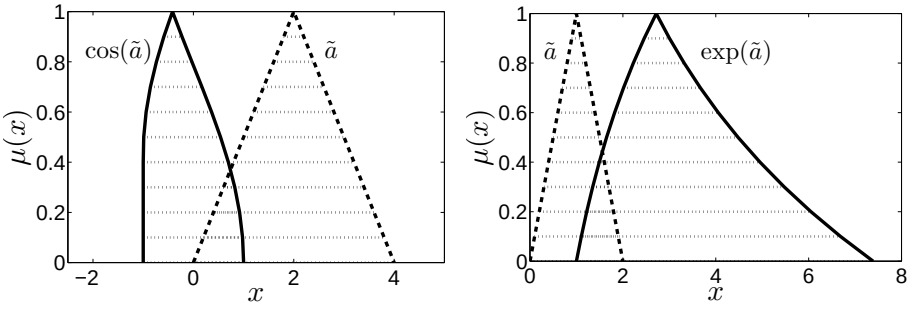
$$[f(\widetilde{a})]_\alpha = f([\widetilde{a}]_\alpha)\,. \tag{2.2}$$

This further simplifies to

$$[f(\widetilde{a})]_\alpha = [\min_{a \in [\widetilde{a}]_\alpha} f(a), \max_{a \in [\widetilde{a}]_\alpha} f(a)] \tag{2.3}$$

when $W = \mathbb{R}$.

Remark. A real-valued function $\mu_{\widetilde{a}}\colon V \to [0,1] \subset \mathbb{R}$ is upper semi-continuous if and only if all of its level sets $\{a \in V : \mu_{\widetilde{a}}(a) \geq \alpha\}$ are closed. Because subsets of a compact set in a metric space are closed if and only if they are



(a) $\cos(\widetilde{a})$ (b) $\exp(\widetilde{a})$

Figure 2.2: Illustration of the computation of the cosine and the exponential of a triangular fuzzy number by the α-cut approach.

compact, the conditions of upper semi-continuity and compact support used in Theorem 2.2.1 are equivalent to the condition that $\widetilde{a}$ is a compact fuzzy set, given that V is a metric space. As such, Theorem 2.2.1 holds, for example, for continuous functions of fuzzy numbers.

Example 2.2.1. *The α-cut approach is illustrated in Fig. 2.2. The picture in the left panel depicts the computation of the cosine of a triangular fuzzy number using evenly distributed α-cuts; the picture in the right panel illustrates the computation of the exponential of a fuzzy number.*

2.3 Interactivity

The extension principle also applies to the fuzzification of functions of several arguments. The simplest examples would be the addition, subtraction, multiplication, and division of fuzzy numbers. The situation is, however, somewhat more complex because the dependency structure between the arguments has to be taken into account. This mutual dependency between two or more fuzzy sets is defined by a joint membership function. In fuzzy set terminology, this dependency is called interactivity.

Definition 2.3.1 (Interactivity, joint membership function)**.** *Let $\widetilde{a} \in \mathcal{F}(V)$ and $\widetilde{b} \in \mathcal{F}(W)$. Then, the interactivity of $\widetilde{a}$ and $\widetilde{b}$ is defined by the joint membership function $\mu_{\widetilde{a},\widetilde{b}}\colon V \times W \to [0,1]$.*

The *marginal* membership function $\mu_{\widetilde{a}}\colon V \to [0,1]$ of $\widetilde{a}$ (and analogously the *marginal* membership function $\mu_{\widetilde{b}}\colon W \to [0,1]$ of $\widetilde{b}$) can be derived from this

joint membership function using Zadeh's extension principle:

$$\mu_{\widetilde{a}}(a) = \sup_{b \in W} \mu_{\widetilde{a},\widetilde{b}}(a,b). \tag{2.4}$$

It is often assumed in fuzzy modeling that the different parameters are noninteractive.

Definition 2.3.2 (Noninteractivity)**.** *Two fuzzy sets $\widetilde{a} \in \mathcal{F}(V)$ and $\widetilde{b} \in \mathcal{F}(W)$ are said to be noninteractive, i.e., independent, if $\mu_{\widetilde{a},\widetilde{b}}(a,b) = \min(\mu_{\widetilde{a}}(a), \mu_{\widetilde{b}}(b))$.*

An important consequence of noninteractivity which we will need later is that when $\widetilde{\xi}_1, \ldots, \widetilde{\xi}_{d_\Xi} \in \mathcal{F}(\mathbb{R})$ are noninteractive fuzzy numbers, the α-cuts of $\widetilde{\boldsymbol{\xi}} := (\widetilde{\xi}_1, \ldots, \widetilde{\xi}_{d_\Xi})$ are hyperrectangles, i.e., $[\widetilde{\boldsymbol{\xi}}]_\alpha = [\widetilde{\xi}_1]_\alpha \times \cdots \times [\widetilde{\xi}_{d_\Xi}]_\alpha$, for all $\alpha \in [0,1]$.

Example 2.3.1. *In the case of noninteractive fuzzy numbers, all computations can be done by means of interval computations with the corresponding α-cuts. An illustration of the procedure for the sum, difference, product and quotient of two noninteractive triangular numbers is provided in Fig. 2.3. The corresponding formula for these operations by the α-cut approach can easily be checked to be*

$$[\widetilde{a} + \widetilde{b}]_\alpha = [[\widetilde{a}]_\alpha^- + [\widetilde{b}]_\alpha^-, [\widetilde{a}]_\alpha^+ + [\widetilde{b}]_\alpha^+],$$

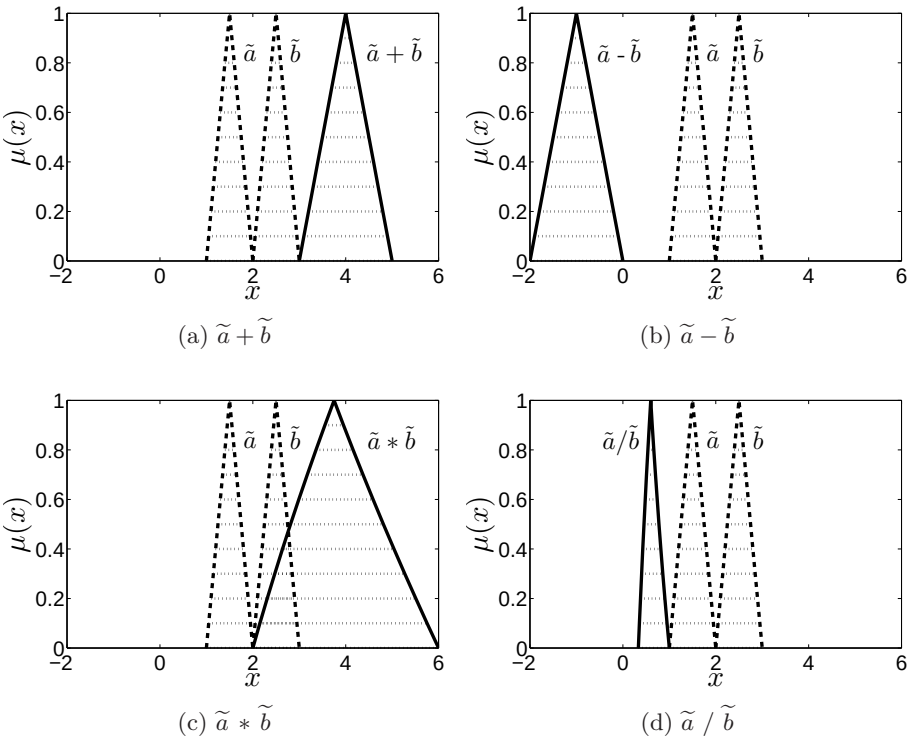
$$[\widetilde{a} - \widetilde{b}]_\alpha = [[\widetilde{a}]_\alpha^- - [\widetilde{b}]_\alpha^+, [\widetilde{a}]_\alpha^+ - [\widetilde{b}]_\alpha^-],$$

$$\begin{aligned}[\widetilde{a} * \widetilde{b}]_\alpha = [&\min\{[\widetilde{a}]_\alpha^-[\widetilde{b}]_\alpha^-, [\widetilde{a}]_\alpha^-[\widetilde{b}]_\alpha^+, [\widetilde{a}]_\alpha^+[\widetilde{b}]_\alpha^-, [\widetilde{a}]_\alpha^+[\widetilde{b}]_\alpha^+\},\\ &\max\{[\widetilde{a}]_\alpha^-[\widetilde{b}]_\alpha^-, [\widetilde{a}]_\alpha^-[\widetilde{b}]_\alpha^+, [\widetilde{a}]_\alpha^+[\widetilde{b}]_\alpha^-, [\widetilde{a}]_\alpha^+[\widetilde{b}]_\alpha^+\}],\end{aligned}$$

$$\begin{aligned}[\widetilde{a} \,/\, \widetilde{b}]_\alpha = [&\min\{[\widetilde{a}]_\alpha^-/[\widetilde{b}]_\alpha^-, [\widetilde{a}]_\alpha^-/[\widetilde{b}]_\alpha^+, [\widetilde{a}]_\alpha^+/[\widetilde{b}]_\alpha^-, [\widetilde{a}]_\alpha^+/[\widetilde{b}]_\alpha^+\},\\ &\max\{[\widetilde{a}]_\alpha^-/[\widetilde{b}]_\alpha^-, [\widetilde{a}]_\alpha^-/[\widetilde{b}]_\alpha^+, [\widetilde{a}]_\alpha^+/[\widetilde{b}]_\alpha^-, [\widetilde{a}]_\alpha^+/[\widetilde{b}]_\alpha^+\}].\end{aligned}$$

2.4 Supremum distance

Important in numerical computations with fuzzy sets, is the notion of distance between fuzzy sets. To that end, we use the so-called supremum distance d_∞ for fuzzy sets [57]. It is based on the Hausdorff distance.



(a) $\widetilde{a}+\widetilde{b}$ (b) $\widetilde{a}-\widetilde{b}$

(c) $\widetilde{a}*\widetilde{b}$ (d) $\widetilde{a}\ /\ \widetilde{b}$

Figure 2.3: Illustration of the sum, difference, product and quotient of two noninteractive triangular numbers.

Definition 2.4.1 (Hausdorff distance)**.** *Let A and B two nonempty subsets of the metric space W with metric d. The Hausdorff distance $d_H(A,B)$ is then defined by*

$$\max\left\{\sup_{a\in A}\inf_{b\in B} d(a,b),\ \sup_{b\in B}\inf_{a\in A} d(a,b)\right\}. \tag{2.5}$$

Definition 2.4.2 (Supremum distance)**.** *Let d_H be the Hausdorff distance on the metric space W. The supremum distance d_∞ on $\mathcal{F}(W)$ is then defined by*

$$d_\infty(\widetilde{a},\widetilde{b}) := \sup_{0\le\alpha\le 1} d_H([\widetilde{a}]_\alpha,[\widetilde{b}]_\alpha)\,. \tag{2.6}$$

For fuzzy numbers $\widetilde{a}$ and $\widetilde{b}$, the Hausdorff distance $d_{\mathrm{H}}([\widetilde{a}]_\alpha,[\widetilde{b}]_\alpha)$ simplifies to $\max\{|[\widetilde{a}]^-_\alpha-[\widetilde{b}]^-_\alpha|,|[\widetilde{a}]^+_\alpha-[\widetilde{b}]^+_\alpha|\}$ if the standard metric for real numbers is used.

Theorem 2.4.1 (Supremum distance for fuzzy numbers)**.** *Let d_H be the Hausdorff distance on $\mathbb{R}$ with the usual metric $d(a,b) = |a-b|$ and let $\widetilde{a}, \widetilde{b} \in \mathcal{F}(\mathbb{R})$ be two fuzzy numbers. We have then that*

$$d_\infty(\widetilde{a}, \widetilde{b}) = \sup_{0 \leq \alpha \leq 1} \max \left\{ |[\widetilde{a}]_\alpha^- - [\widetilde{b}]_\alpha^-|, |[\widetilde{a}]_\alpha^+ - [\widetilde{b}]_\alpha^+| \right\}. \tag{2.7}$$

2.5 Fuzzy sets as imprecise probabilities

There are different views on what the membership degree might represent. Initially, fuzzy sets were introduced to model vague linguistic knowledge in logic [218]. In that regard, fuzzy sets are said to offer a mathematically sound way to operate and reason with imprecise linguistic notions, such as the set of tall persons or the set of interesting papers, or, in case of numerical values, a value which is "about three", or which is "somewhere in between four and five". The membership degree then represents a degree of similarity, of preference, of acceptability, of suitability, etcetera [65, 66]. There is, however, another important interpretation possible, namely that a fuzzy set can be regarded as a credal set, i.e., a set of probability measures. This connection to probability theory is made by interpreting the membership function as a possibility distribution.

Possibility distribution and possibility measure

That the membership function of a fuzzy set can be regarded as a possibility distribution was recognized in [220]. A possibility space is—analogous to a probability space—constructed from a sample space (or universe) V, a set of events $\mathcal{R} \subset 2^V$ (subsets of V are called events), and a possibility measure $\overline{P}\colon \mathcal{R} \to [0,1]$. The possibility space is then conveniently written as the triplet $(V, \mathcal{R}, \overline{P})$. The event space $\mathcal{R}$ is often chosen to be the full power set 2^V, but this is not necessary; see [41, 42, 43, 46] for further details on the definition of possibility spaces with an event space that has the structure of an ample space.

A (normal) possibility measure $\overline{P}$ should satisfy the following axioms:

1. $\overline{P}(\emptyset) = 0$,
2. $\overline{P}(V) = 1$ (normality),

3. for any (possibly uncountable) collection of events $\{E_i \in \mathcal{R}\}_{i\in\mathcal{I}}$:

$$\overline{P}(\bigcup_{i\in\mathcal{I}} E_i) = \sup_{i\in\mathcal{I}} \overline{P}(E_i). \tag{2.8}$$

The third axiom marks the main difference with probability theory. In contrast to probability measures, possibility measures are not additive. In fact, also many of the other existing uncertainty theories can be formalized and characterized using measures. Depending on the specific properties of the measure, one arrives at different theories. See [123] for an overview.

Consider now a fuzzy set $\widetilde{a} \in \mathcal{F}(V)$. The membership function $\mu_{\widetilde{a}}$ can then be used to construct a possibility measure through the definition of the possibility of singleton events $\{a\}$, $a \in V$, as follows:

$$\overline{P}(\{a\}) := \mu_{\widetilde{a}}(a). \tag{2.9}$$

Remark that the third axiom (2.8) combined with (2.9) matches with Zadeh's extension principle (2.1). Let f be a map $f\colon V \to W$ and $(W, \mathcal{R}_W, \overline{P}_W)$ be the possibility space constructed from the membership function of $f(\widetilde{a})$. The possibility of an event $E \in \mathcal{R}_W$ is then equal to the possibility of $f^{-1}(E)$, i.e., $\overline{P}_W(E) = \overline{P}(f^{-1}(E))$.

Necessity measure

Next to the possibility measure in a possibility space, there is also a dual concept, the so-called necessity measure $\underline{P}\colon \mathcal{R} \to [0,1]$. This measure is defined as

$$\underline{P}(E) := 1 - \overline{P}(E^c), \tag{2.10}$$

where $E^c := V \setminus E$ denotes the complement of E. An intuitive interpretation of the two measures is that $\overline{P}(E)$ can be regarded as a measure of the lack of evidence that contradicts the event E and $\underline{P}(E)$ can be regarded as a measure of the evidence that supports the event. In this way, it allows for the modeling of epistemic uncertainties where knowledge is incomplete or imprecise.

Upper and lower probability; Interval probability

The axiom (2.8) is very restrictive, and represents an hypercautious approach to uncertainty. In [207], it is argued that this axiom is actually too restrictive to allow for the modeling of uncertainty in realistic scenarios. When this axiom is dropped and replaced with more general axioms, other more versatile uncertainty models can be defined. The axioms that these theories share are

1. $\overline{P}(\emptyset) = 0$,
2. $\overline{P}(V) = 1$,
3. $\underline{P}(E) = 1 - \overline{P}(E^c)$,
4. $\overline{P}(E) \leq \overline{P}(F)$ for all $E, F \in \mathcal{R}$, $E \subset F$ (monotonicity).

The measures $\overline{P}$ and $\underline{P}$ are then, respectively, referred to as the upper and lower probability. When axioms 1 and 3 are replaced with

1. $\underline{P}(\emptyset) = 0$,
2. $\underline{P}(E) \leq \overline{P}(E)$ for all $E \in \mathcal{R}$,

one arrives at even more general theories. They are collectively called interval probabilities; see [212].

Credal sets

The fact that possibility can be regarded as uncertainty described by an upper and a lower probability, or more generally, an interval probability, provides a very useful interpretation to possibility and fuzzy sets. The two measures $\underline{P}$ and $\overline{P}$ in these theories can be used to define a set of probability measures (i.e., a credal set)

$$\mathcal{M} := \{P : \underline{P}(E) \leq P(E) \leq \overline{P}(E) \text{ for all } E \in \mathcal{R}\}. \tag{2.11}$$

Reversely, an interval probability can be constructed from a credal set $\mathcal{M}$ as

$$\underline{P}(E) := \inf_{P \in \mathcal{M}} P(E), \tag{2.12}$$

$$\overline{P}(E) := \sup_{P \in \mathcal{M}} P(E), \tag{2.13}$$

for all $E \in \mathcal{R}$. If this reverse operation delivers the same interval probability as the one started from, then the interval probability is called an F-probability [212], or equivalently, a coherent interval probability [206, 207]. A coherent interval probability and its corresponding credal set can, as such, be regarded as equivalent. An important consequence is that a *normal* fuzzy set can be interpreted as a credal set, because the interval probability defined by a *normal* possibility measure and its dual necessity measure is known to be coherent; see [44, 45, 205].

Chapter 3

Fuzzy differential equations

This chapter provides the mathematical setting for our study of fuzzy DEs. Such equations have been extensively applied in the engineering literature, see e.g. [3, 51, 149, 151, 160]. As such, we refer to that literature for motivating examples and for a discussion on the modeling aspects. The focus of this chapter will be on the definition and interpretation of fuzzy DEs. The fuzzy diffusion equation is used here as a model equation. We explain how this equation can be discretized such that it can be solved numerically. In Chapter 4, we discuss the computational aspects of solving the large linear algebraic system that results from a Galerkin discretization of the fuzzy diffusion equation. The mathematical framework of the numerical solution to fuzzy DEs as described in this chapter has been published and applied in our papers [36, 38, 39].

3.1 Definition of a fuzzy differential equation and its solution

The first difficulty that arises when dealing with fuzzy DEs is the lack of consensus in the literature on their exact definition. Quite a few alternative (though related) interpretations exist; see, e.g., the discussions in [26, 27, 147, 165, 204]. We will choose the interpretation using sample path-based fuzzy fields [10, 36, 38, 39, 62, 137, 165] and Zadeh's extension principle. Using this approach, the solution to a fuzzy DE will be defined as the fuzzified solution of a corresponding deterministic parametric DE.

3.1.1 Fuzzy processes

Consider the fuzzy ordinary DE (ODE)

$$\frac{\mathrm{d}\widetilde{u}}{\mathrm{d}t}(t) = \widetilde{f}(t, \widetilde{u}(t))\,, \quad \widetilde{u}(0) = \widetilde{u}_0, \tag{3.1}$$

where $t \in [0, \infty)$ represents the time variable, the unknown $\widetilde{u}\colon [0, \infty) \to \mathcal{F}(\mathbb{R})$ is a time dependent fuzzy-valued function (i.e., a fuzzy process), and $\widetilde{f}\colon [0, \infty) \times \mathcal{F}(\mathbb{R}) \to \mathcal{F}(\mathbb{R})$. Trying to interpret and understand this fuzzy DE immediately raises the question of what it means to take the derivative of a fuzzy-valued function.

Looking at the classical definition

$$\frac{\mathrm{d}\widetilde{u}}{\mathrm{d}t}(t) = \lim_{\triangle t \to 0} \frac{\widetilde{u}(t + \triangle t) - \widetilde{u}(t)}{\triangle t} \tag{3.2}$$

and using Zadeh's extension principle to compute $\widetilde{u}(t + \triangle t) - \widetilde{u}(t)$, it becomes clear that knowledge of the joint membership function $\mu_{\widetilde{u}(t+\triangle t), \widetilde{u}(t)}$ on an infinitesimal scale is essential. Assume, for example, that $\widetilde{u}(t) := \widetilde{a}$ for all $t \in [0, \infty)$ with $\widetilde{a}$ some fixed fuzzy number with $[\widetilde{a}]_0^+ - [\widetilde{a}]_0^- > 0$.

- If all $\widetilde{u}(t)$ are assumed to be pairwise noninteractive, then the support of the derivative of the fuzzy-valued function equals

$$[\frac{\mathrm{d}\widetilde{u}}{\mathrm{d}t}(t)]_0 = [-\infty, \infty]. \tag{3.3}$$

- If on the other hand all $\widetilde{u}(t)$ are assumed to be fully interactive, i.e.,

$$\mu_{\widetilde{u}(t_1)}(a) = \mu_{\widetilde{u}(t_2)}(a) \tag{3.4}$$

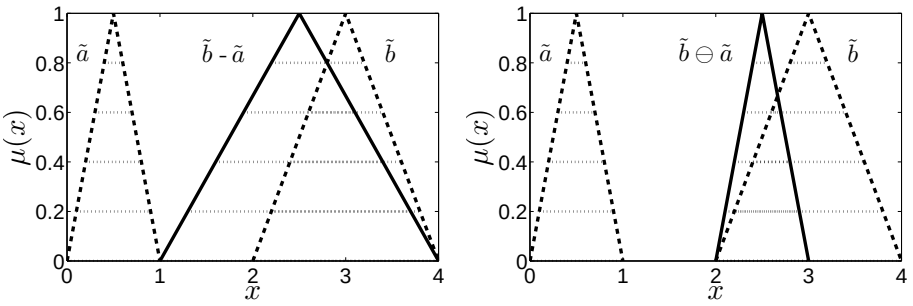
and

$$\mu_{\widetilde{u}(t_1), \widetilde{u}(t_2)}(a, b) = \begin{cases} \mu_{\widetilde{u}(t_1)}(a) & a = b\,, \\ 0 & a \neq b\,, \end{cases} \tag{3.5}$$

then

$$[\frac{\mathrm{d}\widetilde{u}}{\mathrm{d}t}(t)]_0 = [0, 0]. \tag{3.6}$$

These observations for the derivative of a fuzzy-valued function have led to a variety of approaches in the literature for the definition and the interpretation of the derivative of a fuzzy-valued function and fuzzy DEs. The first series of papers dealing with the derivative (and integral) of fuzzy-valued functions seems to be [62, 63, 64]. It is, however, the definition using the Hukuhara difference, first proposed in [175], that attracted more attention.



(a) $\widetilde{b} - \widetilde{a}$ (b) $\widetilde{b} \ominus \widetilde{a}$

Figure 3.1: An illustration of the (standard) difference and the Hukuhara difference of two (noninteractive) fuzzy numbers.

Hukuhara derivative

Let $\widetilde{a}$ and $\widetilde{c}$ be two noninteractive fuzzy numbers and define $\widetilde{b} := \widetilde{c} + \widetilde{a}$ using Zadeh's extension principle for the definition of the addition operation. The fuzzy number $\widetilde{c}$ is then called the Hukuhara difference of $\widetilde{b}$ and $\widetilde{a}$. We denote it by $\widetilde{b} \ominus \widetilde{a}$. Note that the interactivity of $\widetilde{a}$, $\widetilde{b}$, and $\widetilde{c}$ is fixed by the definition of the Hukuhara difference. Necessarily, when computing the Hukuhara difference of two given arbitrary fuzzy numbers, only their marginal membership functions can be used. Any present interactivity information defined by their joint membership function has to be ignored. Note also that the width of the α-cuts of $\widetilde{b}$ needs to be greater than or equal to the width of the corresponding α-cuts of $\widetilde{a}$ in order for $\widetilde{b} \ominus \widetilde{a}$ to exist. See Fig. 3.1 for a comparison of the (standard) difference and the Hukuhara difference of two (noninteractive) fuzzy numbers.

Using the Hukuhara difference, the following derivative can be defined.

Definition 3.1.1 (Hukuhara derivative)**.** *A function* $\widetilde{u}\colon \mathbb{R} \to \mathcal{F}(\mathbb{R})$ *is Hukuhara-differentiable at* $t \in \mathbb{R}$ *if the limits (in the supremum norm topology)*

$$\lim_{\triangle t \to 0^+} \frac{\widetilde{u}(t + \triangle t) \ominus \widetilde{u}(t)}{\triangle t} \quad \textit{and} \quad \lim_{\triangle t \to 0^+} \frac{\widetilde{u}(t) \ominus \widetilde{u}(t - \triangle t)}{\triangle t} \tag{3.7}$$

exist and are equal. The limit is called the Hukuhara derivative.

Some of the first papers that used the Hukuhara derivative or the closely related definition as given in [80] in the study of ordinary differential equations are [109, 110, 185]. A serious downside to the Hukuhara derivative that is mentioned is that it can only exist when the width of the α-cuts of $\widetilde{u}(t)$ is non-decreasing in

time. To overcome this problem, some modifications to the Hukuhara derivative have been proposed; see [14, 15, 25] for further details.

Differential inclusions

In [101], Hüllermeier proposed an entirely different approach based on differential inclusions. It avoids the need to define the derivative. The fuzzy ODE (3.1) is rewritten as a family of differential inclusions

$$\frac{\mathrm{d}u}{\mathrm{d}t}(t) \in [\widetilde{f}(t, u(t))]_\alpha\,, \quad u(0) \in [\widetilde{u}_0]_\alpha, \quad \alpha \in [0,1]. \tag{3.8}$$

The solutions of these inclusions are then used to construct a fuzzy process $\widetilde{u}\colon \mathbb{R} \to \mathcal{F}(\mathbb{R})$ through the definition of its α-cuts

$$[\widetilde{u}(t)]_\alpha = \{u(t)\colon u \text{ is a solution to (3.8)}\}. \tag{3.9}$$

This fuzzy process is called the solution of the fuzzy ODE interpreted as a family of differential inclusions. Further study of this approach to fuzzy ODEs is made in [27, 55, 56, 129].

Sample path-based fuzzy process

The Hukuhara derivative, its variants, and the differential inclusion approach offer a viable definition of fuzzy DEs. They seem, however, somewhat artificial from a possibilistic point of view in the sense that any interactivity information in the fuzzy process is ignored. Arguably a more natural and intuitive way to define a fuzzy process is to define it as a fuzzy set of sample paths, i.e., $\widetilde{u} \in \mathcal{F}(\mathbb{R}^{[0,\infty)})$. The sample paths can then be regarded as realizations of an underlying possibility space. The fuzzy-valued function interpretation of the fuzzy process is easily derived using Zadeh's extension principle:

$$\mu_{\widetilde{u}(t)}(y) = \sup_{u\colon\, y=u(t)} \mu_{\widetilde{u}}(u). \tag{3.10}$$

Also the finite dimensional interactivity in the process follows from Zadeh's extension principle:

$$\mu_{\widetilde{u}(t_1),\ldots,\widetilde{u}(t_n)}(y_1,\ldots,y_n) = \sup_{u\colon\, y_1=u(t_1),\ldots,y_n=(t_n)} \mu_{\widetilde{u}}(u). \tag{3.11}$$

The definition of a fuzzy process as a fuzzy set of sample-paths allows for a straightforward definition of the derivative (and the solution to fuzzy DEs)

through Zadeh's extension principle, i.e.,

$$\mu_{\widetilde{\frac{\mathrm{d}u}{\mathrm{d}t}}}(y) = \sup_{u\colon y=\frac{\mathrm{d}u}{\mathrm{d}t}} \mu_{\widetilde{u}}(u), \tag{3.12}$$

where the derivative could, for example, be defined in the classical sense. The fuzzy set $\widetilde{\frac{\mathrm{d}u}{\mathrm{d}t}}$ thus contains all derivatives of the sample-paths of $\widetilde{u}$. These derivatives are equipped with a membership level computed according to the extension principle.

To the best of our knowledge, the sample path-based definition was first discussed in [62]. The first paper that uses the sample path-based definition to define the derivative and the solution to a fuzzy DE, however, seems to be [137]. A second paper is [165]. Very recently, there has been a rediscovery of the approach applied to the definition of the derivative of a function in [10]. Note that the situation of the dual view on fuzzy processes (i.e. fuzzy-valued function and sample path-based) is analogous to the situation with stochastic processes. There, an equivalence of random-valued functions obeying finite-dimensional joint probability distributions and so-called separable processes (or fields) defined by a probability measure on the set of functions can be proved [2].

3.1.2 Fuzzy fields

An advantage of using the sample path-based approach is that the fuzzy process or field can be defined as a fuzzy set over more useful function spaces than the set of all functions ($\mathbb{R}^{[0,\infty)}$ in the example above), e.g., the space of distributions or some Sobolev space. In the definition below we confine ourselves to fuzzy fields over a spatial domain.

Definition 3.1.2 (Fuzzy field)**.** *A fuzzy field $\widetilde{u}$ over the spatial domain $\Omega \subset \mathbb{R}^{d_\Omega}$ is a fuzzy set over a function space V of functions defined on Ω, i.e., $\widetilde{u} \in \mathcal{F}(V)$.*

Assume, for example, that $\widetilde{u}$ is defined as a fuzzy set of distributions on $\Omega \subset \mathbb{R}^{d_\Omega}$. The gradient $\nabla\widetilde{u}$ is then easily defined using Zadeh's extension principle:

$$\mu_{\widetilde{\nabla u}}(\boldsymbol{y}) = \sup_{u\colon \boldsymbol{y}=\nabla u} \mu_{\widetilde{u}}(u) \tag{3.13}$$

with the gradient of u defined in the distributional sense. This shows that fuzzy fields defined as such allow for a natural study of fuzzy differential equations. Moreover, the norm of a normed function space can be straightforwardly used to define a distance between sample path-based fuzzy fields; see Definition 2.4.2.

In Section 3.2.3, we derive a useful and computable upper bound to the supremum distance.

As a final note, we would like to mention that the sample path-based approach to fuzzy DEs is widely used in the literature but without any explicit discussion of the sample path-based definition of fuzzy fields and the derivative. The solution to the fuzzy DE is straightly defined as the fuzzification by Zadeh's extension principle of the deterministic solution. Fuzzy finite-dimensional parameters in $\mathcal{F}(\mathbb{R}^d)$ are propagated through the (discretized) solution operator of the corresponding deterministic DE in analogy to the method used in [165]. Papers that adhere to this strategy are, for example, [26, 147, 204]. When a finite element discretization is used, this approach is sometimes referred to as the fuzzy finite element method [1, 3, 29, 150, 152, 151, 148, 177, 203].

3.1.3 Example problem: the Brusselator

As a first example of a fuzzy DE, we take the Brusselator. It is a model for a type of autocatalytic chemical reaction. The reaction scheme of the Brusselator is

$$A \xrightarrow{k_1} X,$$

$$2X + Y \xrightarrow{k_2} 3X,$$

$$B + X \xrightarrow{k_3} Y + D,$$

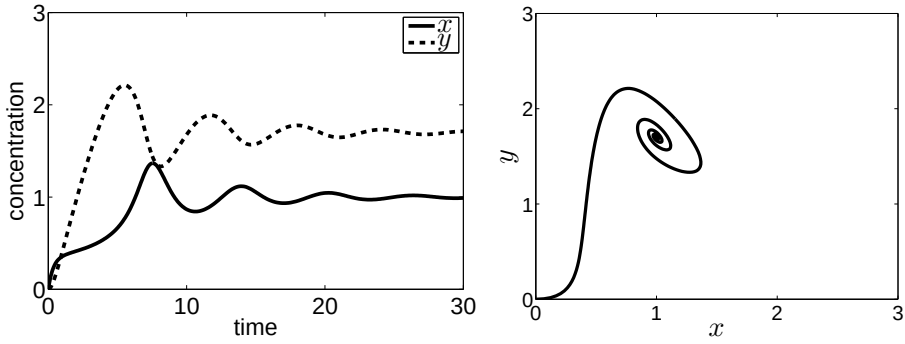
$$X \xrightarrow{k_4} E,$$

where A, B, D, E, X, Y are the reagents, and $k_1, \ldots, k_4$ are the reaction rates. For simplicity, we assume $k_1 = \cdots = k_4 = 1$. Further, we denote the molar concentrations of A, B, D, E, X, Y, respectively, by a, b, d, e, x, y, and assume a, b, d, e to be constants. The molar concentrations x and y are then governed by the following system of ODEs:

$$\begin{cases} \dfrac{\mathrm{d}x}{\mathrm{d}t}(t,a,b) = a + x(t,a,b)^2\, y(t,a,b) - b\, x(t,a,b) - x(t,a,b), \\ \dfrac{\mathrm{d}y}{\mathrm{d}t}(t,a,b) = b\, x(t,a,b) - x(t,a,b)^2\, y(t,a,b), \end{cases} \tag{3.14}$$

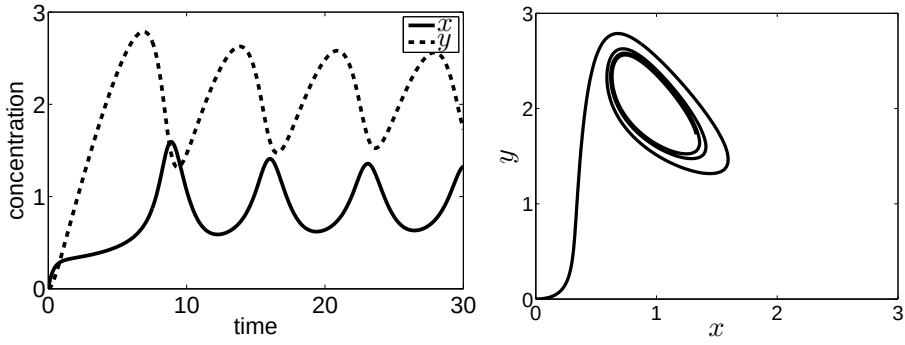
where we explicitly wrote x and y as dependent on a and b, as they will be used as parameters of the system. We supplement (3.14) with the initial conditions

$$\frac{\mathrm{d}x}{\mathrm{d}t}(0,a,b) = \frac{\mathrm{d}y}{\mathrm{d}t}(0,a,b) = 0. \tag{3.15}$$



(a) (b)

Figure 3.2: The Brusselator (3.14) in a stable regime with $a = 1$ and $b = 1.7$.



(a) (b)

Figure 3.3: The Brusselator (3.14) in an unstable regime with $a = 0.9$ and $b = 1.9$.

This dynamical system has a fixed point at $x = a$, $y = b/a$, which is stable if $b < 1 + a^2$. Taking, for example, $a = 1$ and $b = 1.7$, the concentrations x and y are attracted in a spiral to the point $(1, 1.7)$; see Fig. 3.2. For $a = 0.9$ and $b = 1.9$, the Brusselator approaches a limit cycle; see Fig. 3.3.

We will now introduce some fuzziness to the Brusselator. We replace a and b with noninteractive triangular fuzzy numbers $\widetilde{a}$ and $\widetilde{b}$ that have a support $[\widetilde{a}]_0 = [0.9, 1.1]$ and $[\widetilde{b}]_0 = [1.5, 1.9]$. Consequently, the solution of the system

of ODEs will also be fuzzy. The system becomes

$$\begin{cases} \dfrac{\mathrm{d}x}{\mathrm{d}t}(t,\widetilde{a},\widetilde{b}) = \widetilde{a} + x(t,\widetilde{a},\widetilde{b})^2\, y(t,\widetilde{a},\widetilde{b}) - \widetilde{b}\, x(t,\widetilde{a},\widetilde{b}) - x(t,\widetilde{a},\widetilde{b}), \\ \dfrac{\mathrm{d}y}{\mathrm{d}t}(t,\widetilde{a},\widetilde{b}) = \widetilde{b}\, x(t,\widetilde{a},\widetilde{b}) - x(t,\widetilde{a},\widetilde{b})^2\, y(t,\widetilde{a},\widetilde{b}). \end{cases} \tag{3.16}$$

Following the interpretation of fuzzy DEs by sample path-based fuzzy fields and Zadeh's extension principle, the concentrations $x(\cdot,\widetilde{a},\widetilde{b})$ and $y(\cdot,\widetilde{a},\widetilde{b})$ can be obtained through the fuzzification by Zadeh's extension principle of the corresponding deterministic problem (3.14). Their membership functions are then equal to

$$\mu_{x(\cdot,\widetilde{a},\widetilde{b})}(z) = \sup_{a,b:\ \begin{cases} z=x(\cdot,a,b) \\ \text{eq. (3.14)} \end{cases}} \mu_{\widetilde{a},\widetilde{b}}(a,b), \tag{3.17}$$

$$\mu_{y(\cdot,\widetilde{a},\widetilde{b})}(z) = \sup_{a,b:\ \begin{cases} z=y(\cdot,a,b) \\ \text{eq. (3.14)} \end{cases}} \mu_{\widetilde{a},\widetilde{b}}(a,b), \tag{3.18}$$

where due to the noninteractivity of $\widetilde{a}$ and $\widetilde{b}$, we have by Definition 2.3.2 that

$$\mu_{\widetilde{a},\widetilde{b}}(a,b) = \min(\mu_{\widetilde{a}}(a), \mu_{\widetilde{b}}(b)). \tag{3.19}$$

Further assuming that all solutions $x(\cdot,a,b)$, $y(\cdot,a,b)$, $a \in [\widetilde{a}]_0$, $b \in [\widetilde{b}]_0$ are classical solutions, we can evaluate them pointwise, such that

$$\mu_{x(t,\widetilde{a},\widetilde{b})}(z) = \sup_{a,b:\ \begin{cases} z=x(t,a,b) \\ \text{eq. (3.14)} \end{cases}} \min(\mu_{\widetilde{a}}(a), \mu_{\widetilde{b}}(b)), \tag{3.20}$$

$$\mu_{y(t,\widetilde{a},\widetilde{b})}(z) = \sup_{a,b:\ \begin{cases} z=y(t,a,b) \\ \text{eq. (3.14)} \end{cases}} \min(\mu_{\widetilde{a}}(a), \mu_{\widetilde{b}}(b)). \tag{3.21}$$

These membership functions $\mu_{x(t,\widetilde{a},\widetilde{b})}$ and $\mu_{y(t,\widetilde{a},\widetilde{b})}$ are very hard to compute directly. We will instead use the α-cut approach. Note, however, that Theorem 2.2.1 (i.e., the α-cut approach) can only be applied if the mappings $(a,b) \mapsto x(t,a,b)$ and $(a,b) \mapsto y(t,a,b)$ are continuous over $[(\widetilde{a},\widetilde{b})]_0$. This is something we will not prove here, but simply assume. We have then that

$$[x(t,\widetilde{a},\widetilde{b})]_\alpha = [\min_{a\in[\widetilde{a}]_\alpha, b\in[\widetilde{b}]_\alpha} x(t,a,b), \max_{a\in[\widetilde{a}]_\alpha, b\in[\widetilde{b}]_\alpha} x(t,a,b)], \tag{3.22}$$

$$[y(t,\widetilde{a},\widetilde{b})]_\alpha = [\min_{a\in[\widetilde{a}]_\alpha, b\in[\widetilde{b}]_\alpha} y(t,a,b), \max_{a\in[\widetilde{a}]_\alpha, b\in[\widetilde{b}]_\alpha} y(t,a,b)] \tag{3.23}$$

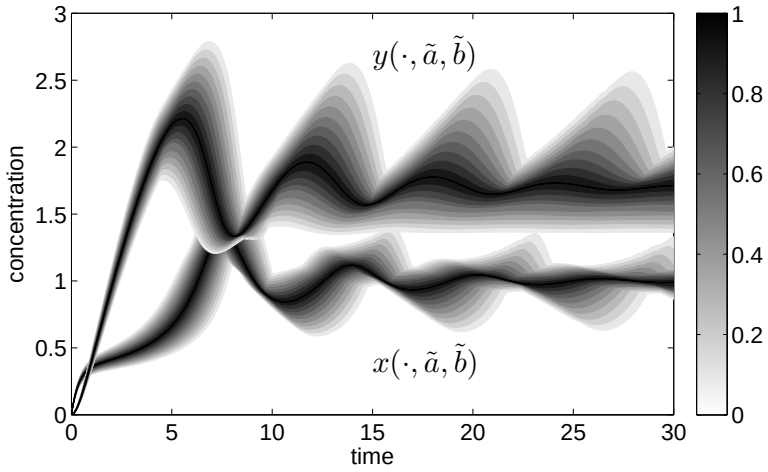


Figure 3.4: The solution of the fuzzy Brusselator (3.16) with $\widetilde{a}$ and $\widetilde{b}$ noninteractive symmetrical triangular fuzzy numbers that have a support $[\widetilde{a}]_0 = [0.9, 1.1]$ and $[\widetilde{b}]_0 = [1.5, 1.9]$. The gray-scale colors represent the membership values $\mu_{x(t,\widetilde{a},\widetilde{b})}$ and $\mu_{y(t,\widetilde{a},\widetilde{b})}$.

with $x(\cdot, a, b)$ and $y(\cdot, a, b)$ satisfying (3.14).

Computing some of the α-cuts of $x(t, \widetilde{a}, \widetilde{b})$ and $(t, \widetilde{a}, \widetilde{b})$ for different $t \in [0, \infty)$ then amounts to solving many ODE-constrained optimization problems. They share much information though. We solve the problem here by solving (3.14) (with a standard ODE solver in Matlab) for values of (a, b) that are obtained from an equidistant tensor grid sampling of $[\widetilde{a}]_0 \times [\widetilde{b}]_0$. These solutions are then evaluated at the time values $t_1, \ldots, t_n$ for which we wish to know the fuzzy values $x(t, \widetilde{a}, \widetilde{b})$ and $y(t, \widetilde{a}, \widetilde{b})$. The optimization problems in (3.22)–(3.23) are solved by simply selecting the minimal and maximal values of the tensor grid samples. Fig. 3.4 shows the result of this computation using a tensor grid sampling with 41×41 samples, $\alpha = 0, 0.1, \ldots, 1$, and 400 equidistant time points in $[0, 30]$.

The remainder of this thesis will mainly focus on some numerical techniques to do such computations more efficiently and accurately (with the error measured in the supremum distance for fuzzy sets). As model problem, we will not use the Brusselator, but instead use the steady-state diffusion equation (see next section). The techniques that we will investigate in the following chapters can be roughly categorized as:

- Chapter 3, Chapter 4, Chapter 6: the construction of a response surface approximation to the parametric problem (in the case of the Brusselator (3.16), this would amount to constructing a numerical approximation of x and y over the time domain as well as the parameter domain, i.e., $[0,t]\times[\widetilde{a}]_0\times[\widetilde{b}]_0$) which can then be used for the computation of the α-cuts of different kinds of quantities of interest (such as the pointwise values).
- Chapter 5: derivative-free global optimization technique for the computation of the α-cuts with a focus on high-dimensionality.

3.2 The fuzzy elliptic model problem

3.2.1 A fuzzy partial differential equation

We consider an elliptic PDE defined on a d_Ω-dimensional Lipschitz domain $\Omega \subset \mathbb{R}^{d_\Omega}$ with a fuzzy diffusion coefficient. This equation will be written as

$$-\nabla\cdot(\widetilde{a}\,\nabla\widetilde{u}) = f, \qquad \widetilde{u}|_{\partial\Omega} = 0, \tag{3.24}$$

with $f \in H^{-1}(\Omega)$ the deterministic source term, $\widetilde{a} \in \mathcal{F}(L^\infty(\Omega))$ the fuzzy input field, and $\widetilde{u} \in \mathcal{F}(H_0^1(\Omega))$ the unknown fuzzy solution field. Finally, in order to ensure the problem is well-posed, we assume that there exist real numbers $a_{\min}$ and $a_{\max}$ for which

$$0 < a_{\min} \leq a(\boldsymbol{x}) \leq a_{\max} < \infty \tag{3.25}$$

almost everywhere for all $a \in [\widetilde{a}]_0$.

Following the interpretation of fuzzy PDEs by sample path-based fuzzy fields and Zadeh's extension principle, the fuzzy PDE is defined as the fuzzification of the parametric PDE

$$-\nabla\cdot\big(a\,\nabla u(\cdot,a)\big) = f, \qquad u(\cdot,a)|_{\partial\Omega} = 0, \tag{3.26}$$

where $a \in [\widetilde{a}]_0$ is a particular realization of the fuzzy field, and acting now as a parameter. Function $u(\cdot,a) \in H_0^1(\Omega)$ is the corresponding solution parameterized by a. Using the solution operator

$$S\colon L^\infty(\Omega) \supset [\widetilde{a}]_0 \to H_0^1(\Omega)\colon a \mapsto u(\cdot,a)\,, \tag{3.27}$$

this translates to the following.

Definition 3.2.1 (Solution to fuzzy PDE (3.24))**.** *The solution $\widetilde{u}$ to the fuzzy PDE* (3.24) *is the fuzzy field obtained through fuzzification of the solution operator S of the corresponding parametric PDE* (3.26)*, i.e., $\widetilde{u} = S(\widetilde{a})$, with $\widetilde{a}$ the fuzzy input field.*

The fuzzy field $\widetilde{u}$ thus contains all possible realizations of the form $S(a)$ with $a \in [\widetilde{a}]_0$, and each of those realizations is equipped with a membership level computed according to the extension principle (2.1).

3.2.2 Model for the fuzzy diffusion coefficient

We will now introduce a modeling assumption for the fuzzy input field $\widetilde{a}$. It resembles the typical representation of stochastic fields by a Karhunen–Loève expansion [77]. That is, we consider a fuzzy input field that is (or has been approximated by) an expansion of the form

$$\widetilde{a} := a_0 + \sigma \sum_{k=1}^{d_\Xi} a_k \widetilde{\xi}_k \,, \tag{3.28}$$

where the $\widetilde{\xi}_1, \ldots, \widetilde{\xi}_{d_\Xi}$ are assumed to be noninteractive fuzzy numbers with support $[-1, 1]$. As such, the fuzzy parameter $\widetilde{\boldsymbol{\xi}} := (\widetilde{\xi}_1, \ldots, \widetilde{\xi}_{d_\Xi})$ takes values in $\Xi := [\widetilde{\boldsymbol{\xi}}]_0 = [-1, 1]^{d_\Xi}$. This space Ξ will be referred to as the parameter domain or the uncertainty domain. We include both the finite-dimensional case, i.e., $d_\Xi < \infty$, and the infinite-dimensional case, i.e., $d_\Xi = \infty$. The factor σ is introduced for convenience of easy scaling of the fuzzy part. Function $a_0 \in L^\infty(\Omega)$ represents the main component of the fuzzy field $\widetilde{a}$, while the other terms with $a_k \in L^\infty(\Omega)$, $k = 1, \ldots, d_\Xi$, express the fuzzy variations of the field. Convergence of the deterministic equivalent of the series is assumed to be in $C(\Xi; L^\infty(\Omega))$, i.e.,

$$a := a_0 + \sigma \sum_{k=1}^{d_\Xi} a_k \xi_k \quad \in C(\Xi; L^\infty(\Omega)). \tag{3.29}$$

Using the above model (3.28) for the diffusion coefficient, the parameterized representation of (3.24) can be written as

$$-\nabla \cdot \big(a(\cdot, \boldsymbol{\xi}) \nabla u(\cdot, \boldsymbol{\xi})\big) = f, \qquad u(\cdot, \boldsymbol{\xi})|_{\partial\Omega} = 0. \tag{3.30}$$

Assuming again that there exist real numbers $a_{\min}$ and $a_{\max}$ for which

$$0 < a_{\min} \leq a(\boldsymbol{x}, \boldsymbol{\xi}) \leq a_{\max} < \infty \tag{3.31}$$

almost everywhere in Ω and for all $\boldsymbol{\xi} \in [\widetilde{\boldsymbol{\xi}}]_0$, the Lax-Milgram theorem ensures the well-posedness of the PDE (3.30) for all $\boldsymbol{\xi} \in [\widetilde{\boldsymbol{\xi}}]_0$. As such we can again define a solution operator

$$S\colon \Xi \to H_0^1(\Omega)\colon \boldsymbol{\xi} \mapsto u(\cdot, \boldsymbol{\xi}). \tag{3.32}$$

This map has a bounded analytic extension to an open set which is strictly larger than Ξ; see [34, Lemma 2.2]. As such we have

$$S \in C(\Xi; H_0^1(\Omega)). \tag{3.33}$$

Remark. The proposed model for the diffusion coefficient, which resembles the Karhunen–Loève expansion of a stochastic field, may seem somewhat artificial in a fuzzy context. The modeling of fuzzy fields, or, more generally, epistemically uncertain fields, is, however, still a topic of ongoing research; see e.g. [148, 153, 203] and the references therein. As such, we do not claim that this model is the only possible approach. Alternatives, such as polynomial representations resembling polynomial chaos type techniques, can certainly also be considered in the modeling of fuzzy fields in actual applications.

3.2.3 Supremum distance for fuzzy fields

In order to measure the accuracy of a numerical approximation to the exact solution $\widetilde{u}$ of (3.24), we use the supremum distance (2.6). We recall the definition of the supremum distance and the Hausdorff distance here for the reader's convenience: let $\widetilde{u}$, $\widetilde{v} \in \mathcal{F}(W)$, where W is a metric space with metric d, then

$$d_\infty(\widetilde{u}, \widetilde{v}) = \sup_{0 \le \alpha \le 1} d_{\mathrm{H}}([\widetilde{u}]_\alpha, [\widetilde{v}]_\alpha) \tag{3.34}$$

with

$$d_{\mathrm{H}}([\widetilde{u}]_\alpha, [\widetilde{v}]_\alpha) = \max\left\{ \sup_{u \in [\widetilde{u}]_\alpha} \inf_{v \in [\widetilde{v}]_\alpha} d(u,v), \sup_{v \in [\widetilde{v}]_\alpha} \inf_{u \in [\widetilde{u}]_\alpha} d(u,v) \right\}. \tag{3.35}$$

Consider the case where we have a numerical approximation $\widetilde{v} := v(\cdot, \widetilde{\boldsymbol{\xi}})$ of the exact solution $\widetilde{u} := u(\cdot, \widetilde{\boldsymbol{\xi}})$ of our model problem (3.24). The natural metric d to be used in the Hausdorff distance would then be the one that is derived

from the $H_0^1(\Omega)$-norm. As such, we should be able to compute

$$\max\left\{\sup_{\boldsymbol{\xi}_1\in[\widetilde{\boldsymbol{\xi}}]_\alpha}\inf_{\boldsymbol{\xi}_2\in[\widetilde{\boldsymbol{\xi}}]_\alpha}\|u(\cdot,\boldsymbol{\xi}_1)-v(\cdot,\boldsymbol{\xi}_2)\|_{H_0^1(\Omega)},\right.$$
$$\left.\sup_{\boldsymbol{\xi}_2\in[\widetilde{\boldsymbol{\xi}}]_\alpha}\inf_{\boldsymbol{\xi}_1\in[\widetilde{\xi}]_\alpha}\|u(\cdot,\boldsymbol{\xi}_1)-v(\cdot,\boldsymbol{\xi}_2)\|_{H_0^1(\Omega)}\right\} \tag{3.36}$$

to know the supremum distance $d_\infty(\widetilde{u},\widetilde{v})$. Obviously, this is a very hard problem. We can, however, derive the following upper bound.

Theorem 3.2.1. *Let $\widetilde{\xi}\in\mathcal{F}(V)$ be a fuzzy set with compact support and upper semi-continuous membership function over some topological space V, and W a metric space with metric d. If $u\colon[\widetilde{\xi}]_0\to W$ and $v\colon[\widetilde{\xi}]_0\to W$ are continuous maps, then*

$$d_\infty(\widetilde{u},\widetilde{v})\le\sup_{\xi\in[\widetilde{\xi}]_0}d(u(\xi),v(\xi))\,, \tag{3.37}$$

where $\widetilde{u}:=u(\widetilde{\xi})$ and $\widetilde{v}:=v(\widetilde{\xi})$.

Proof. Combining (3.34) with Theorem 2.2.1, we get

$$d_\infty(u(\widetilde{\xi}),v(\widetilde{\xi}))=\sup_{0\le\alpha\le1}d_{\mathrm{H}}(u([\widetilde{\xi}]_\alpha),v([\widetilde{\xi}]_\alpha))\,. \tag{3.38}$$

By definition of the Hausdorff distance (3.35), $d_{\mathrm{H}}(u([\widetilde{\xi}]_\alpha),v([\widetilde{\xi}]_\alpha))$ is equal to

$$\max\left\{\sup_{\xi_1\in[\widetilde{\xi}]_\alpha}\inf_{\xi_2\in[\widetilde{\xi}]_\alpha}d(u(\xi_1),v(\xi_2)),\ \sup_{\xi_2\in[\widetilde{\xi}]_\alpha}\inf_{\xi_1\in[\widetilde{\xi}]_\alpha}d(u(\xi_1),v(\xi_2))\right\}.$$

This can be bounded as

$$d_{\mathrm{H}}(u([\widetilde{\xi}]_\alpha),v([\widetilde{\xi}]_\alpha))\le\sup_{\xi\in[\widetilde{\xi}]_\alpha}d(u(\xi),v(\xi))\le\sup_{\xi\in[\widetilde{\xi}]_0}d(u(\xi),v(\xi))\,.$$

Together with (3.38), this completes the proof. □

Corollary 3.2.1. *Under the same assumptions as in Theorem 3.2.1, and with W being a normed vector space, we have*

$$d_\infty(\widetilde{u},\widetilde{v})\le\|u-v\|_{C([\widetilde{\xi}]_0;W)}\,. \tag{3.39}$$

Remark. In the case of continuous functions $[\widetilde{\xi}]_0 \to W$ over a compact set $[\widetilde{\xi}]_0 \subset \mathbb{R}^{d_\Xi}$ with $d_\Xi < \infty$, the $C([\widetilde{\xi}]_0; W)$-norm is equal to the $L^\infty([\widetilde{\xi}]_0; W)$-norm, i.e., the supremum is the same as the essential supremum. As such, Corollary 3.2.1 is in line with the existing literature about epistemic modeling [67, 78, 103] which states that accuracy in the L^∞-norm is required in case of epistemic uncertainty.

Corollary 3.2.2. *Under the assumption that the series* (3.29) *converges in* $C(\Xi; L^\infty(\Omega))$*, it follows that*

$$a_0 + \sigma \sum_{k=1}^{d_\Xi} a_k \widetilde{\xi}_k \tag{3.40}$$

converges in $\mathcal{F}(L^\infty(\Omega))$ *w.r.t. the supremum distance.*

3.3 Numerical approximation by a response surface method

3.3.1 A response surface approach

The approach to "solving" a fuzzy DE depends on what the quantities of interest are. The pointwise value $\widetilde{u}(\boldsymbol{x})$ of the fuzzy field at some point $\boldsymbol{x} \in \Omega$ or the average value $\frac{1}{|\Omega|}\int_\Omega \widetilde{u}(\boldsymbol{x})\mathrm{d}(\boldsymbol{x})$ are examples of typical quantities of interest. In these cases, the α-cut approach described in Theorem 2.2.1 can be applied using the solution operator S (3.33), because these quantities of interest are continuous functions over Ξ. Remark, here, that in case of pointwise evaluation, sufficient regularity of f is needed (i.e., such that the elliptic regularity theorem can be applied).

The α-cut approach amounts to a PDE-constrained minimization and maximization problem for each α-level considered. For example, in the case of a pointwise evaluation, this formula becomes

$$[\widetilde{u}(\boldsymbol{x})]_\alpha = [\min_{\boldsymbol{\xi} \in [\widetilde{\boldsymbol{\xi}}]_\alpha} u(\boldsymbol{x}, \boldsymbol{\xi}), \max_{\boldsymbol{\xi} \in [\widetilde{\boldsymbol{\xi}}]_\alpha} u(\boldsymbol{x}, \boldsymbol{\xi})] \tag{3.41}$$

with $u(\cdot, \boldsymbol{\xi})$ satisfying

$$-\nabla \cdot \big(a(\cdot, \boldsymbol{\xi})\, \nabla u(\cdot, \boldsymbol{\xi})\big) = f, \qquad u(\cdot, \boldsymbol{\xi})|_{\partial\Omega} = 0. \tag{3.42}$$

Moreover, the procedure would have to be repeated for each new quantity of interest considered, for example, for each new evaluation point $\boldsymbol{x}$.

In order to reduce the computational cost, it is therefore common practice to construct a numerical approximation u^{r}—also referred to as a response surface approximation—of the solution u of the parametric PDE (3.30) during a preprocessing stage [3, 51, 120]. Such an approximation u^{r} typically takes the form of a sum of separable functions, i.e.,

$$u^{\mathrm{r}}(\boldsymbol{x}, \boldsymbol{\xi}) = \sum_{j=1}^{n_{\mathrm{r}}} u_j(\boldsymbol{x}) r_j(\boldsymbol{\xi}) \tag{3.43}$$

with cheap to evaluate continuous basis functions $(r_j)_{j=1}^{n_{\mathrm{r}}}$ and coefficients $(u_j)_{j=1}^{n_{\mathrm{r}}}$. In the next sections, we will discuss a few of the numerical techniques to construct such a response surface approximation.

Once constructed, the response surface can be used to cheaply compute all sorts of quantities of interest. Assume, for example, for our model problem (3.24), that we would like to compute a quantity of interest that is represented by a continuous linear functional $q \in H^{-1}(\Omega)$. Special case here is, for example, the weighted average operator. Using the α-cut approach (Theorem 2.2.1) and Corollary 3.2.1, we have then that

$$[q(\widetilde{u}^{\mathrm{r}})]_\alpha = [\min_{\boldsymbol{\xi} \in [\widetilde{\boldsymbol{\xi}}]_\alpha} q(u^{\mathrm{r}}(\cdot, \boldsymbol{\xi})), \max_{\boldsymbol{\xi} \in [\widetilde{\boldsymbol{\xi}}]_\alpha} q(u^{\mathrm{r}}(\cdot, \boldsymbol{\xi}))] \tag{3.44}$$

with an approximation error in the fuzzy supremum distance that can be bounded as

$$d_\infty(q(\widetilde{u})), q(\widetilde{u}^{\mathrm{r}})) \leq \|q(u(\cdot, \boldsymbol{\xi})) - q(u^{\mathrm{r}}(\cdot, \boldsymbol{\xi}))\|_{C(\Xi)} \tag{3.45}$$

$$\leq \|q\|_{H^{-1}(\Omega)} \|u - u^{\mathrm{r}}\|_{C(\Xi; H_0^1(\Omega))} \,. \tag{3.46}$$

The accuracy of such a quantity of interest is thus guaranteed if u^{r} is accurate in the $C(\Xi; H_0^1(\Omega))$-norm.

3.3.2 Approximation by polynomials

The main computational challenge when trying to solve (3.30) and construct a numerical approximation to u is the large (possibly countably infinite) dimension of $\Omega \times \Xi$. Already for a moderate d_Ξ, it is important to carefully choose a numerical method for the construction of an approximation to u, due to the so-called curse of dimensionality [16, 188].

The fact that the mapping $\boldsymbol{\xi} \to u(\cdot, \boldsymbol{\xi})$ is analytic in an open set strictly larger than Ξ, makes polynomial approximation in Ξ an obvious choice, i.e.,

we construct an approximation of the form

$$u^{\mathrm{r}}(\cdot,\boldsymbol{\xi}) = \sum_{\boldsymbol{j}\in\mathcal{J}} u_{\boldsymbol{j}} \prod_{k=1}^{d_\Xi} \xi_k^{j_k}, \tag{3.47}$$

where $\mathcal{J} \subset \mathcal{F}_{d_\Xi}$ with

$$\mathcal{F}_{d_\Xi} := \left\{ \boldsymbol{i} \in \mathbb{N}_0^{d_\Xi} \colon \|\boldsymbol{i}\|_0 < \infty \right\} \tag{3.48}$$

the set of d_Ξ-dimensional finitely supported multi-indices, and where coefficients $u_{\boldsymbol{j}} \in H_0^1(\Omega)$.

An important result about the approximability of elements in $C(\Xi; H_0^1(\Omega))$ by polynomials of the form (3.47) can be found in [78, Theorem 3.2]. We repeat it here. For any finite set $\mathcal{J} \subset \mathcal{F}_{d_\Xi}$, let

$$\mathbb{P}_{\mathcal{J}} := \operatorname{span}\left\{ \prod_{k=1}^{d_\Xi} \xi_k^{j_k} : \boldsymbol{j} \in \mathcal{J} \right\}, \tag{3.49}$$

and

$$\mathbb{P}(\Xi; X) := \bigcup_{\mathcal{J}\subset\mathcal{F}_{d_\Xi}} X \otimes \mathbb{P}_{\mathcal{J}}, \tag{3.50}$$

i.e., the vector space of polynomials over Ξ with coefficients in X. Then for any Banach space X with the approximation property (e.g., a separable Hilbert space), we have the following density result.

Theorem 3.3.1. [78, Theorem 3.2]. *If X is a Banach space with the approximation property, then $\mathbb{P}(\Xi; X)$ is dense in $C(\Xi; X)$.*

This is a generalization of the Stone-Weierstrass theorem to Banach space-valued functions.

For the concrete case of the approximability of the analytic solution $u \in C(\Xi; H_0^1(\Omega))$ of (3.30), we recall the following result from [34].

Theorem 3.3.2. [34, Theorem 3.2]. *Let $a \in C(\Xi; L^\infty(\Omega))$ be defined as in* (3.29) *and satisfy* (3.31). *If $\|(\|a_k\|_{L^\infty(\Omega)})_{k=1}^{d_\Xi}\|_p < \infty$ for some $0 < p < 1$, then for $u \in C(\Xi; H_0^1(\Omega))$, the solution of* (3.30), *there exists a sequence $\mathcal{J}_1 \subset \mathcal{J}_2 \subset \cdots \subset \mathcal{F}_{d_\Xi}$ with $\#(\mathcal{J}_N) = N$ for which*

$$\inf_{v\in H_0^1(\Omega)\otimes\mathbb{P}_{\mathcal{J}_N}} \|u - v\|_{C(\Xi;H_0^1(\Omega))} = \mathcal{O}(N^{-s}), \quad N \to \infty \tag{3.51}$$

with $s := \frac{1}{p} - 1$.

As shown in [30], the index sets $\mathcal{J}_N$ in Theorem 3.3.2 can be chosen from the class of downward closed index sets

Definition 3.3.1 (Downward closed index set)**.** *An index set $\mathcal{J} \subset \mathcal{F}_{d_\Xi}$ is called downward closed if*

$$\boldsymbol{j} \in \mathcal{J} \;\; \textit{and} \;\; \boldsymbol{i} \leq \boldsymbol{j} \;\Rightarrow\; \boldsymbol{i} \in \mathcal{J}. \tag{3.52}$$

Hence, it is theoretically possible to achieve an algebraic convergence rate when $d_\Xi = \infty$ and a faster than algebraic convergence rate when $d_\Xi < \infty$ with sparse polynomial approximations. Of course, the questions arise now how to choose $\mathcal{J}_N$ and how to construct a good polynomial $u^{\mathrm{r}} \in H_0^1(\Omega) \otimes \mathbb{P}_{\mathcal{J}_N}$.

Remark. The choice for polynomial approximation is evident in this case. The parametric PDE at hand may, however, not always have such nice smoothness properties. In the literature on stochastic and parametric PDEs, various approaches have been suggested to deal with problems exhibiting a less smooth dependence on the parameters. One can, for example, find methods based on wavelets [134, 135], finite elements [5, 6, 120], multi-elements [74, 104, 141, 209], splines [165], as well as response surface methods based on Kriging, radial basis functions, regression models, etcetera [3, 51, 67].

3.3.3 Discretization of the fuzzy and deterministic dimensions

The choice of appropriate polynomial spaces $\mathbb{P}_{\mathcal{J}_N}$ for the discretization of the fuzzy dimensions can be based on a priori estimates [17, 31, 33, 34, 75, 163, 194] or in an adaptive way on a posteriori estimates [6, 17, 31, 75, 76, 78, 79, 121, 163]. Then, to compute a $u^{\mathrm{r}} \in H_0^1(\Omega) \otimes \mathbb{P}_{\mathcal{J}_N}$, one can choose several strategies to discretize the PDE (3.30) into a finite-dimensional algebraic problem. The most popular of such methods for the discretization in the parameter domain Ξ are the collocation approach [4, 31, 76, 121, 157, 162, 163, 194, 215] (which happens to be equivalent to interpolation in the Ξ space) and the Galerkin approach [5, 6, 75, 79, 144, 194, 216]. Other methods include the pseudo-spectral method [21, 35, 214] and the Neumann series expansion method [78]. For some of the connections between the different methods, we refer to [7, 20, 35]. In §3.3.4 and §3.3.5 below, we provide further details on the collocation and the Galerkin approach.

Apart from the discretization in the parameter domain Ξ, the equation should also be discretized in the spatial domain Ω, i.e., the deterministic dimensions. This is typically done using a Galerkin projection on a standard finite element space defined on a triangulation of Ω, or by means of a finite difference or finite volume method. Reduced basis methods could be applied here to reduce

the dimension of the spatial approximation space [18, 86, 146]. Both the discretization in the parameter domain and the discretization in the spatial domain can be combined in several ways, leading to a variety of solution strategies. This can complicate the error analysis considerably. To keep things concise, we will restrict ourselves to a discussion of the error introduced by a discretization in the parameter domain only. For an example of an adaptive method which balances both the spatial and the parameter discretization error, and which controls the error in the $C(\Xi; H_0^1(\Omega))$-norm, we refer to [78].

The most common approach to deal with the possible infinite-dimensionality of the parameter domain of the PDE (3.30) is to first truncate the series in the diffusion coefficient (3.29) by retaining only the important, i.e., large terms. This is the so-called finite dimensional noise assumption. Say the series was truncated to the first d_Ξ^{tr} terms and define a^{tr} as the truncated series, $\Xi^{\mathrm{tr}} := [-1,1]^{d_\Xi^{\mathrm{tr}}}$, and u^{tr} as the corresponding solution. Let u^{r} be a numerical approximation to u^{tr}. The total error of this approximation can then be estimated as

$$\|u - u^{\mathrm{r}}\|_{C(\Xi;H_0^1(\Omega))} \leq \|u - u^{\mathrm{tr}}\|_{C(\Xi;H_0^1(\Omega))} + \|u^{\mathrm{tr}} - u^{\mathrm{r}}\|_{C(\Xi^{\mathrm{tr}};H_0^1(\Omega))}. \tag{3.53}$$

Hence, when aiming for an accurate and efficient solution of the problem, these two error terms should be carefully balanced. To this end, the error estimate

$$\|u - u^{\mathrm{tr}}\|_{C(\Xi;H_0^1(\Omega))} \leq \frac{\|f\|_{H^{-1}(\Omega)}}{a_{\min}} \|a - a^{\mathrm{tr}}\|_{C(\Xi;H_0^1(\Omega))}, \tag{3.54}$$

of the first term could be useful. It is a direct corollary of [34, Lemma 2.1].

3.3.4 Spectral collocation method

The most straightforward way to discretize the parametric PDE (3.30) in the parameter domain is the collocation method. Collocation in the parameter domain leads to an interpolation problem when constructing the response surface. As such it is nonintrusive, i.e., standard solvers and software can be used without any modification to solve the PDE in the spatial domain for designated values of the parameters.

Let $S_N := \{\boldsymbol{s}_i\}_{i=1}^N \subset \Xi$ be a set of interpolation nodes and Y_N an N-dimensional linear subspace of $\mathbb{P}(\Xi;\mathbb{R})$. Lagrangian interpolation of a function $u \in C(\Xi; X)$ with X some Banach space is then defined as: find a function $u^{\mathrm{r}} \in X \otimes Y_N$ such that

$$u^{\mathrm{r}}(\cdot, \boldsymbol{s}_i) = u(\cdot, \boldsymbol{s}_i), \quad i = 1, \ldots, N. \tag{3.55}$$

This problem does not always have a solution and, in the case there is a solution, it is not always unique. Given that the problem is well-posed, we can assign an interpolation operator $I_N \colon C(\Xi; X) \to X \otimes Y_N \subset C(\Xi; X)$ to it. Using the Lebesgue theorem, the error in the $C(\Xi; X)$-norm can then bounded as

$$\|u - I_N u\|_{C(\Xi;X)} \leq (1 + \|I_N\|) \inf_{v \in X \otimes Y_N} \|u - v\|_{C(\Xi;X)}, \tag{3.56}$$

where

$$\|I_N\| := \sup_{v \neq 0} \frac{\|I_N v\|_{C(\Xi;X)}}{\|v\|_{C(\Xi;X)}} \tag{3.57}$$

is the so-called Lebesgue constant. The question can then be raised which sets of interpolation nodes S_N and which approximation spaces Y_N yield a small value of $\|I_N\|$ and a small value of $\inf_{v \in X \otimes Y_N} \|u - v\|_{C(\Xi;X)}$.

Univariate interpolation

First, we discuss the univariate case. The theory for univariate Lagrangian interpolation is well developed and the construction of sets of interpolation nodes and approximation spaces in the multivariate case is often derived from the univariate case. For the results we discuss here, we refer to the book [196].

Let

$$S_n := \{s_{i,n}\}_{i=0}^n \subset [-1, 1] \tag{3.58}$$

be a set of $n + 1$ distinct interpolation nodes and $\mathbb{P}_n := \mathbb{P}_{\{0,\ldots,n\}} \equiv \operatorname{span}\{\xi^0, \ldots, \xi^n\}$ the set of univariate polynomials of maximal degree n. The corresponding Lagrangian interpolation operator, which we denote by I_n, is an operator $I_n \colon C([-1, 1], X) \to X \otimes \mathbb{P}_n \subset C([-1, 1], X)$, which defines a unique interpolation $I_n u$ of a function $u \in C([-1, 1]; X)$ that is equal to

$$I_n u = \sum_{i=0}^{n} u(\cdot, s_{i,n}) l_{i,n}, \tag{3.59}$$

where

$$l_{i,n}(\xi) := \prod_{j=0, j \neq i}^{n} \frac{\xi - s_{j,n}}{s_{i,n} - s_{j,n}} \tag{3.60}$$

are the Lagrange polynomials.

The Lebesgue constant $\|I_n\|$ is equal to

$$\|I_n\| = \sup_{\xi\in[-1,1]} \sum_{i=0}^{n} |l_{i,n}(\xi)|. \tag{3.61}$$

Regardless of the choice of nodes, it can be shown that

$$\|I_n\| \geq \frac{2}{\pi}\log(n+1) + 0.52125\ldots. \tag{3.62}$$

For the particular choice of the extrema

$$s_{i,n} = \cos(i\pi/n), \quad i = 0,\ldots,n, \tag{3.63}$$

of the Chebyshev polynomials (see Section 3.3.5 for a definition of the Chebyshev polynomials) as interpolation nodes—also called the Chebyshev–Gauss–Lobatto nodes—we have a close to optimal value of $\|I_n\|$, namely

$$\frac{2}{\pi}\log(n+1) + 0.9625\ldots \leq \|I_n\| \leq \frac{2}{\pi}\log(n+1) + 1. \tag{3.64}$$

We will refer to these nodes as the Chebyshev nodes. An infamous example of a bad set of interpolation nodes is the set of equidistant nodes $s_{i,n} = -1 + 2(i-1)/(n-1)$, $i = 0,\ldots,n$. For these nodes, one has

$$\|I_n\| \geq \frac{2^{n-2}}{n^2}. \tag{3.65}$$

As mentioned before, polynomial approximation achieves very high rates of convergence if the function possesses some smoothness. More particularly, we have exponential convergence

$$\inf_{v\in X\otimes\mathbb{P}_n} \|u - v\|_{C([-1,1];X)} \leq C\rho^{-n} \tag{3.66}$$

for some $\rho > 1$ and $C > 0$, if $u \in C([-1,1],X)$ can be analytically extended to an open set strictly larger than $[-1,1]$. For a proof in the case of Banach space-valued functions, see [4, Lemma 4.4]. In combination with the bound (3.64) for the Lebesgue constant and using the Lebesgue theorem (3.56), this results in

$$\|u - I_n u\|_{C([-1,1],X)} \leq \left(\frac{2}{\pi}\log(n+1) + 2\right) C\rho^{-n}. \tag{3.67}$$

for Lagrangian interpolation in the Chebyshev nodes.

Tensor grid interpolation

Sets of multivariate interpolation nodes can be constructed from the sets of univariate interpolation nodes $S_n := \{s_{i,n}\}_{i=0}^{n} \subset [-1,1]$, by taking the Cartesian product, i.e., for $\boldsymbol{n} \in \mathcal{F}_{d_\Xi}$ we define

$$S_{\boldsymbol{n}} := \bigtimes_{k=1}^{d_\Xi} S_{n_k}. \tag{3.68}$$

The corresponding interpolation operator is defined as

$$I_{\boldsymbol{n}} := \bigotimes_{k=1}^{d_\Xi} I_{n_k}. \tag{3.69}$$

It is a map $I_{\boldsymbol{n}} \colon C(\Xi, X) \to X \otimes \mathbb{P}_{\boldsymbol{n}} \subset C(\Xi, X)$, where

$$\mathbb{P}_{\boldsymbol{n}} := \bigotimes_{k=1}^{d_\Xi} \mathbb{P}_{n_k}. \tag{3.70}$$

The Lebesgue constant of this operator can be shown (see [31]) to be

$$\|I_{\boldsymbol{n}}\| = \prod_{k=1}^{d_\Xi} \|I_{n_k}\|. \tag{3.71}$$

In the simple case that $d_\Xi < \infty$ and $n := n_1 = \cdots = n_{d_\Xi}$, we have again the same exponential convergence

$$\inf_{v \in X \otimes \mathbb{P}_{\boldsymbol{n}}} \|u - v\|_{C(\Xi;X)} \leq C\rho^{-n} \tag{3.72}$$

as in the univariate case for some $\rho > 1$ and $C > 0$, under the assumption that the function $u \in C(\Xi, X)$ can be analytically extended to an open set strictly larger than Ξ. Note, however, that the total number of interpolation nodes is $N = n^{d_\Xi}$. Therefore, as a function of N, we get

$$\inf_{v \in X \otimes \mathbb{P}_{\boldsymbol{n}}} \|u - v\|_{C(\Xi;X)} \leq C\rho^{-N^{\frac{1}{d_\Xi}}}. \tag{3.73}$$

Here, we can observe a manifestation of the curse of dimensionality.

The curse of dimensionality can be partially alleviated by using an anisotropic tensor grid with the n_k, $k = 1, \ldots, d_\Xi$, tuned to the size of the domain of analyticity in each coordinate ξ_k. See [6] for an adaptive strategy based on a posteriori error estimation.

Another approach to handle the exponential increase of complexity with the dimension, which has recently found considerable attention in the literature, is based on low-rank tensor methods [8, 60, 70, 114, 115, 113, 126, 125, 145, 167]. We will use this technology in Chapter 5 for the fuzzification of a real-valued function by the α-cut approach (e.g., the computation of quantities of interest of $u^{\mathrm{r}}(\cdot, \widetilde{\boldsymbol{\xi}})$).

Sparse grid interpolation

In short, sparse grid interpolation is interpolation on a grid which is defined as a *union* of tensor grids (3.68). For simplicity of notation and exposition, we confine ourselves here to the case where the sets S_n of interpolation nodes are nested.

Let $(s_i)_{i\geq 0}$ be a sequence of mutually distinct points in $[-1, 1]$ and let

$$S_n := \{s_i\}_{i=0}^{n}. \tag{3.74}$$

For a downward closed index set $\mathcal{J} \subset \mathcal{F}_{d_\Xi}$, we define the union of Cartesian product grids

$$S_{\mathcal{J}} := \bigcup_{\boldsymbol{j}\in\mathcal{J}} S_{\boldsymbol{j}}. \tag{3.75}$$

For nested sets S_n of interpolation nodes, this is equivalent to

$$S_{\mathcal{J}} = \{\boldsymbol{s_i} \colon \boldsymbol{i} \in \mathcal{J}\}, \tag{3.76}$$

where

$$\boldsymbol{s_i} := (s_{i_1}, \ldots, s_{i_{d_\Xi}}). \tag{3.77}$$

Further, we define the operators

$$\Delta_n := I_n - I_{n-1} \quad \text{and} \quad \Delta_{\boldsymbol{n}} := \bigotimes_{k=1}^{d_\Xi} \Delta_{n_k}. \tag{3.78}$$

The interpolation operator corresponding to the grid $S_{\mathcal{J}}$ can then be defined by the Smolyak construction [190] as

$$I_{\mathcal{J}} := \sum_{\boldsymbol{j}\in\mathcal{J}} \Delta_{\boldsymbol{j}}. \tag{3.79}$$

By [31, Theorem 2.1], it is a map $I_{\mathcal{J}}\colon C(\Xi, X) \to X \otimes \mathbb{P}_{\mathcal{J}} \subset C(\Xi, X)$ which interpolates functions $u \in C(\Xi, X)$ on the grid $S_{\mathcal{J}}$.

A crude upper bound to the Lebesgue constant of the operator $I_{\mathcal{J}}$ is given by [31, Lemma 3.1].

Theorem 3.3.3. [31, Lemma 3.1]. *If the Lebesgue constants of I_n satisfy*

$$\|I_n\| \leq (n+1)^{\theta}, \quad n \geq 0 \tag{3.80}$$

for some $\theta \geq 1$, then the Lebesgue constant of $I_{\mathcal{J}}$ satisfies

$$\|I_{\mathcal{J}}\| \leq (\#(\mathcal{J}))^{\theta+1} \tag{3.81}$$

for any downward closed index set $\mathcal{J}$.

An adaptive method to build the index set $\mathcal{J}$ and which uses a Leja sequence of interpolation nodes $(s_i)_{i\geq 0}$ can be found in [31]. The most common sparse grid interpolation methods found in the literature, however, use Chebyshev nodes and impose a specific structure on the index set [7, 11, 76, 121, 162, 163, 215]. In the notation used above, they can be defined as follows.

First, we define the sets S_n as in (3.58) containing the Chebyshev nodes (3.63). From this sequence of sets, we retain only the sets S_{2^i}, $i \geq 0$. These sets are nested. As such it is possible to construct a sequence $(s_i)_{i\geq 0}$ of mutually distinct points from the nodes in these sets such that the corresponding sequence of sets S_n, $n \geq 0$, as defined in (3.74), has the same sets S_{2^i}, $i \geq 0$, defined before. Let

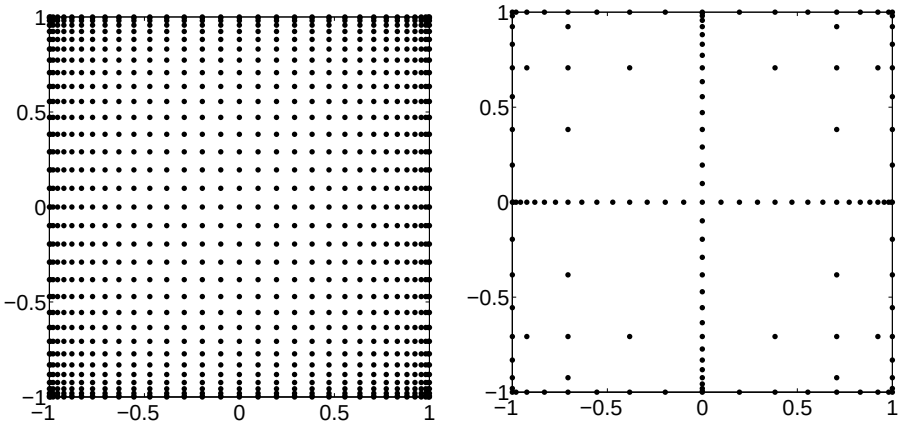
$$\mathcal{I}_n^0 := \{0, \ldots, n\}, \quad \mathcal{I}_{\boldsymbol{n}}^0 := \bigtimes_{k=1}^{d_\Xi} \mathcal{I}_{n_k}^0, \tag{3.82}$$

and define the anisotropic index set $\mathcal{J}_{\boldsymbol{w},p}^{\mathrm{SM}} \subset \mathcal{F}_{d_\Xi}$ as

$$\mathcal{J}_{\boldsymbol{w},p}^{\mathrm{SM}} := \bigcup_{\substack{\boldsymbol{j}:=(2^{i_k})_{k=0}^{d_\Xi} \\ \boldsymbol{i}\in\mathcal{F}_{d_\Xi}\colon \boldsymbol{w}\cdot\boldsymbol{i}\leq p}} \mathcal{I}_{\boldsymbol{j}}^0, \tag{3.83}$$

for some integer $p \geq 0$ and weight $\boldsymbol{w} \in \mathbb{R}^{d_\Xi}$ with $\inf_{k=1,\ldots,d_\Xi} w_k \geq 0$. An illustration of a tensor grid and a sparse grid is given in Fig. 3.5.

The Lagrangian interpolation $\mathcal{I}_{\mathcal{J}_{\boldsymbol{w},p}^{\mathrm{SM}}} u$ of the solution u of (3.30) for both the case $d_\Xi < \infty$ and $d_\Xi = \infty$ can be proven to converge at an algebraic rate if the weight $\boldsymbol{w}$ is properly adapted to the domain of analyticity and if there is a sufficient increase of the size of the domain of analyticity in the coordinate ξ_k for $k \to \infty$. For more details we refer to [163]. We recall [163, Theorem 3.8] here in a simplified and modified version using [4, Lemma 3.2] and adapted to our mathematical setting.



(a) tensor grid $S_{(2^5,2^5)}$ (#nodes = 1089) (b) sparse grid $S_{\mathcal{J}^{\mathrm{SM}}_{(1,1),5}}$ (#nodes = 145)

Figure 3.5: Illustration of an isotropic tensor grid and an isotropic sparse grid based on Chebyshev nodes.

Theorem 3.3.4. [163, Theorem 3.8]. *Let* $a \in C(\Xi; L^\infty(\Omega))$ *be defined as in* (3.29) *and satisfy* (3.31). *Further, let the weight* $\boldsymbol{w} \in \mathbb{R}^{d_\Xi}$ *be defined by*

$$w_k = \frac{1}{2}\log\left(\frac{a_{min}}{2\|a_k\|_{L^\infty(\Omega)}} + \sqrt{1 + \frac{a_{min}^2}{4\|a_k\|^2_{L^\infty(\Omega)}}}\right), \quad k = 1, \ldots, d_\Xi.$$

Then, for $u \in C(\Xi; H_0^1(\Omega))$, *the solution of* (3.30), *it holds that*

$$\|u - I_{\mathcal{J}^{SM}_{\boldsymbol{w},p}} u\|_{C(\Xi;X)} \leq CN^{-s} \tag{3.84}$$

for some $C > 0$, *with* $N = \#(\mathcal{J}^{SM}_{\boldsymbol{w},p})$ *the number of interpolation nodes, and*

$$s = \frac{\log(2)e - 1/2}{\log(2)/\inf_{k=1,\ldots,d_\Xi}(w_k) + \sum_{k=1}^{d_\Xi} 1/w_k}.$$

Remark. This theorem is based on an a priori definition of the weight $\boldsymbol{w}$. An adaptive strategy based on a posteriori estimates can also be found in [163].

3.3.5 Spectral Galerkin method

The Galerkin discretization of the parametric PDE (3.30) in the parameter domain starts from the semi-weak formulation which reads as follows: find

$u \in H_0^1(\Omega) \otimes L_w^2(\Xi)$ such that, for all $v \in L_w^2(\Xi)$,

$$-\int_\Xi \nabla \cdot \big(a(\cdot,\boldsymbol{\xi})\nabla u(\cdot,\boldsymbol{\xi})\big)\, v(\boldsymbol{\xi}) w(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi} = \int_\Xi f v(\boldsymbol{\xi}) w(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}. \tag{3.85}$$

This semi-weak formulation is then discretized by restricting $u(\boldsymbol{x},\cdot)$ and v to be an element of some finite-dimensional subspace of $L_w^2(\Xi)$. We choose the (normalized) Chebyshev weight function

$$w(\boldsymbol{\xi}) = \prod_{k=1}^{d_\Xi} \frac{1}{\pi\sqrt{1-\xi_k^2}}. \tag{3.86}$$

as weight function and $\mathbb{P}_\mathcal{J} \subset L_w^2(\Xi)$ with $\mathcal{J} \subset \mathcal{F}_{d_\Xi}$ some downward closed index set as finite-dimensional approximation space.

The choice of the Chebyshev weight is motivated by the observation that a L_w^2 projection, with Chebyshev weight w, of a continuous function on a polynomial approximation space results in a quasi-optimal approximation in the $C(\Xi;X)$-norm [143, 196]. A common other choice in the case of epistemic uncertainty is the Legendre weight $w(\boldsymbol{\xi}) = 1$ [103, 210]. The performance in the $C(\Xi;X)$-norm of polynomial approximation with both the Chebyshev and the Legendre weight is very similar [22].

An orthonormal basis for $\mathbb{P}_\mathcal{J}$ w.r.t. the Chebyshev weight is provided by the (normalized) multivariate Chebyshev polynomials $T_{\boldsymbol{j}}(\boldsymbol{\xi}) := T_{j_1}(\xi_1)\cdots T_{j_{d_\Xi}}(\xi_{d_\Xi})$, $\boldsymbol{j} \in \mathcal{J}$, where

$$T_0(\xi) := 1\,, \quad T_j(\xi) := \sqrt{2}\cos(j\arccos(\xi))\,, \quad j = 1,2,\ldots \tag{3.87}$$

are the (normalized) univariate Chebyshev polynomials. For notational convenience, we reorder the multivariate Chebyshev polynomials for $\mathbb{P}_\mathcal{J}$ into a single sequence of basis functions $\psi_1,\ldots,\psi_{n_\psi}$, where $n_\psi = \#(\mathcal{J})$. The discrete form of (3.85) then reads: find

$$u_{n_\psi}^{\mathrm{r}} = \sum_{j=1}^{n_\psi} u_j \psi_j \ \in H_0^1(\Omega) \otimes \mathbb{P}_\mathcal{J} \tag{3.88}$$

with coefficients $u_j \in H_0^1(\Omega)$ such that, for all $v \in \mathbb{P}_\mathcal{J}$, equation (3.85) holds.

Together with the model (3.28) for the diffusion coefficient a, this leads to a coupled system of PDEs:

$$-\left(\boldsymbol{G}_0\left(\nabla\cdot(a_0\nabla\cdot)\right) + \sum_{k=1}^{d_\Xi} \boldsymbol{G}_k\left(\nabla\cdot(\sigma a_k \nabla\cdot)\right)\right)\boldsymbol{u} = \boldsymbol{g}\, f, \tag{3.89}$$

where $\boldsymbol{G}_0$ equals the identity matrix $\boldsymbol{I}_{n_\psi}$,

$$\boldsymbol{G}_{k|i,j} = \int_\Xi \xi_k \psi_i(\boldsymbol{\xi}) \psi_j(\boldsymbol{\xi}) w(\boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\xi}, \qquad i,j = 1,\dots,n_\psi, \quad k = 1,\dots,d_\Xi, \tag{3.90}$$

and

$$g_j = \int_\Xi \psi_j(\boldsymbol{\xi}) w(\boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\xi}, \qquad j = 1,\dots,n_\psi. \tag{3.91}$$

The vector $\boldsymbol{u}$ contains the coefficients u_j in (3.88) columnwise.

The most common polynomial approximation spaces that are used in the literature are either (anisotropic) tensor product spaces [5, 6, 75], i.e., $\mathbb{P}_{\mathcal{J}^{\mathrm{TP}}_{\boldsymbol{w},p}}$ with

$$\mathcal{J}^{\mathrm{TP}}_{\boldsymbol{w},p} := \left\{ \boldsymbol{j} \in \mathcal{F}_{d_\Xi} \colon \max_{k=1,\dots,d_\Xi} j_k w_k \leq p \right\}, \tag{3.92}$$

or sparse spaces containing polynomials up to a certain (weighted) total degree [144, 216], i.e., $\mathbb{P}_{\mathcal{J}^{\mathrm{TD}}_{\boldsymbol{w},p}}$ with

$$\mathcal{J}^{\mathrm{TD}}_{\boldsymbol{w},p} := \left\{ \boldsymbol{j} \in \mathcal{F}_{d_\Xi} \colon \boldsymbol{j} \cdot \boldsymbol{w} \leq p \right\}, \tag{3.93}$$

where $p \geq 0$ is some integer and $\boldsymbol{w} \in \mathbb{R}^{d_\Xi}$ the weight. Despite the possible anisotropy of the approximation space with tensor product structure, its size can still grow very fast. A reduction of the computational work could then be achieved by using low-rank tensor methods; see the discussion about low-rank tensor based methods for tensor grid interpolation in Section 3.3.4. For a further comparison of the polynomial spaces used in the collocation approach as discussed in Section 3.3.4 and the Galerkin approach discussed here, see [7].

3.3.6 The fully discrete problem

The discretization in the parameter space by the spectral collocation or the spectral Galerkin approach as discussed in previous sections can be combined in several ways with a discretization in the spatial domain; see the discussion in Section 3.3.3. For both the collocation and the Galerkin approach, we will use a finite-dimensional approximation space $X_h \subset H^1_0(\Omega)$ of standard Lagrange finite elements on a triangulation of Ω for the discretization in the spatial domain. Further, let $\{\phi_i\}_{i=1}^{n_\phi}$ be a basis for X_h.

Collocation approach

We arrive at a fully discrete problem in the collocation case by starting with a Galerkin discretization of the parametric PDE (3.30) in the spatial domain. Such a Galerkin discretization is derived from the following parametric semi-weak formulation: find $u(\cdot,\boldsymbol{\xi}) \in H_0^1(\Omega)$ such that, for all $v \in H_0^1(\Omega)$,

$$\int_\Omega a(\boldsymbol{x},\boldsymbol{\xi})\nabla u(\boldsymbol{x},\boldsymbol{\xi}) \cdot \nabla v(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} = \int_\Omega f(\boldsymbol{x})v(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x}. \tag{3.94}$$

Using the approximation space $X_h \subset H_0^1(\Omega)$, the discrete form of (3.94) then reads: find

$$u_{n_\phi}^{\mathrm{r}}(\cdot,\boldsymbol{\xi}) = \sum_{i=1}^{n_\phi} u_i(\boldsymbol{\xi})\phi_i \ \in X_h \tag{3.95}$$

with coefficients $u_i \colon \Xi \to \mathbb{R}$ such that, for all $v \in X_h$, equation (3.94) holds.

Together with the model (3.28) for the diffusion coefficient a, this leads to the parametric linear algebraic system

$$\left(\boldsymbol{A}_0 + \sum_{k=1}^{d_\Xi} \boldsymbol{A}_k \,\xi_k\right) \boldsymbol{u}(\boldsymbol{\xi}) = \boldsymbol{f}, \tag{3.96}$$

where

$$\boldsymbol{A}_{0|i,j} = \int_\Omega a_0(\boldsymbol{x})\nabla\phi_i(\boldsymbol{x}) \cdot \nabla\phi_j(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x}, \qquad i,j = 1,\dots,n_\phi, \tag{3.97}$$

$$\boldsymbol{A}_{k|i,j} = \int_\Omega \sigma a_k(\boldsymbol{x})\nabla\phi_i(\boldsymbol{x}) \cdot \nabla\phi_j(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x}, \qquad i,j = 1,\dots,n_\phi, \tag{3.98}$$

$$k = 1,\dots,d_\Xi,$$

$$f_i = \int_\Omega f(\boldsymbol{x})\phi_i(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x}, \qquad i = 1,\dots,n_\phi. \tag{3.99}$$

The vector $\boldsymbol{u}(\boldsymbol{\xi})$ contains the coefficients $u_i(\boldsymbol{\xi})$ in (3.95) columnwise.

Let $I_{\mathcal{J}} \colon C(\Xi, H_0^1(\Omega)) \to H_0^1(\Omega) \otimes \mathbb{P}_{\mathcal{J}} \subset C(\Xi, H_0^1(\Omega))$ be a Lagrangian interpolation operator with $\mathcal{J} \subset \mathcal{F}_{d_\Xi}$ some downward closed index. Further, let $\psi_1,\dots,\psi_{n_\psi}$ be a basis for $\mathbb{P}_{\mathcal{J}}$, where $n_\psi = \#(\mathcal{J})$. Then, we have that $I_{\mathcal{J}} u_{n_\phi}^{\mathrm{r}}$ can be written as

$$u_{n_\phi,n_\psi}^{\mathrm{r}} = \sum_{i=1}^{n_\phi}\sum_{j=1}^{n_\psi} u_{i,j}\phi_i\psi_j \ \in X_h \otimes \mathbb{P}_{\mathcal{J}} \tag{3.100}$$

with coefficients $u_{i,j} \in \mathbb{R}$.

Galerkin approach

To arrive at a fully discrete problem in the Galerkin case, we apply a Galerkin discretization procedure in both the parameter and the spatial domain. The fully weak formulation of the parametric PDE (3.30) reads as follows: find $u \in H_0^1(\Omega) \otimes L_w^2(\Xi)$ such that, for all $v \in H_0^1(\Omega) \otimes L_w^2(\Xi)$,

$$\int_\Xi \int_\Omega a(\boldsymbol{x}, \boldsymbol{\xi}) \nabla u(\boldsymbol{x}, \boldsymbol{\xi}) \cdot \nabla v(\boldsymbol{x}, \boldsymbol{\xi}) w(\boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}\boldsymbol{\xi} = \int_\Xi \int_\Omega f(\boldsymbol{x}) v(\boldsymbol{\xi}) w(\boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}\boldsymbol{\xi}, \quad (3.101)$$

where w is the Chebyshev weight function (3.86).

Using the approximation spaces $X_h \subset H_0^1(\Omega)$ and $\mathbb{P}_\mathcal{J} \subset L_w^2(\Xi)$ with $\mathcal{J} \subset \mathcal{F}_{d_\Xi}$ some downward closed index set, the discrete form of (3.101) then reads: find

$$u_{n_\phi, n_\psi}^{\mathrm{r}} = \sum_{i=1}^{n_\phi} \sum_{j=1}^{n_\psi} u_{i,j} \phi_i \psi_j \in X_h \otimes \mathbb{P}_\mathcal{J} \quad (3.102)$$

with coefficients $u_{i,j} \in \mathbb{R}$ and $\{\psi_j\}_{j=1}^{n_\psi}$ an orthonormal basis for $\mathbb{P}_\mathcal{J}$, provided by the (normalized) multivariate Chebyshev polynomials, such that, for all $v \in X_h \otimes \mathbb{P}_\mathcal{J}$, equation (3.101) holds.

Together with the model (3.28) for the diffusion coefficient a, this leads to the linear algebraic system

$$\left(\boldsymbol{G}_0 \otimes \boldsymbol{A}_0 + \sum_{k=1}^{d_\Xi} \boldsymbol{G}_k \otimes \boldsymbol{A}_k \right) \boldsymbol{u} = \boldsymbol{g} \otimes \boldsymbol{f} \quad (3.103)$$

with $\boldsymbol{G}_0$ the identity matrix $\boldsymbol{I}_{n_\psi}$, and $\boldsymbol{G}_k$, $\boldsymbol{g}$, $\boldsymbol{A}_0$, $\boldsymbol{A}_k$, $\boldsymbol{f}$ as defined in, respectively, (3.90), (3.91), (3.97), (3.98), (3.99). The vector $\boldsymbol{u}$ contains the coefficients $u_{i,j}$ in (3.102) columnwise.

3.4 Conclusions

We started this chapter with an overview of some of the possible interpretations and definitions of fuzzy DEs. Two of the main approaches that can be found in the literature are the one based on the Hukuhara derivative and some of its variants, and the one based on differential inclusions. These approaches seem,

however, somewhat artificial in the light of the possibilistic interpretation of fuzzy sets. Arguably a more natural and intuitive approach to fuzzy DEs is then to define the fuzzy processes and fields in these DEs as fuzzy sets of sample paths. Fuzzy processes or fields can be defined as fuzzy sets over the space of distributions or Sobolev spaces, allowing for a very natural study of fuzzy differential equations. The solution to the fuzzy DE is simply defined as the fuzzification by Zadeh's extension principle of the solution to the corresponding parametric DE.

As a first illustration of the sample path-based interpretation, we discussed the Brusselator problem in which some of the parameters are assumed to be fuzzy. Trying to compute the pointwise value of the solution by the α-cut approach showed us that we had to solve a long sequence of similar ODE-constrained optimization problems. We addressed this problem straightforwardly by solving the deterministic ODE for a large number of different values of the parameters. The optimization problems were then solved by optimizing over these precomputed deterministic solutions. This brought us to make a precise statement of the objectives of the rest of this thesis, namely developing numerical procedures for a more accurate and efficient solution of the optimization problems that arise in the solution process of a fuzzy DE.

We continued this chapter with a discussion of the main model problem used in this thesis: the diffusion problem with a fuzzy diffusion coefficient that is modeled in a way similar to the Karhunen–Loève expansion of stochastic fields. We discussed the response surface approach to solving fuzzy DEs and showed that if the response surface is accurate in the $C([\widetilde{\xi}]_0; X)$-norm, then the fuzzy quantities of interest computed from this response surface are accurate in the supremum distance d_∞ for fuzzy sets. The chapter continued with an overview of several numerical methods to construct an accurate (in the $C([\widetilde{\xi}]_0; X)$-norm) spectral polynomial response surface approximation. We focused in particular on the collocation and the Galerkin approach, for which we gave a summary of some of the error estimates that can be found in the literature. The remainder of this thesis will now focus on the efficient solution of the large algebraic system which results from the Galerkin discretization (Chapter 4), a derivative-free global optimization technique for the computation of the α-cuts based on low-rank tensor methods (Chapter 5), and recycling techniques for the efficient computation of long sequences of similar systems which arise, for example, in the collocation approach (Chapter 6).

Chapter 4

Preconditioners for the Galerkin system

This chapter focuses on the computational aspects of solving the large linear system (3.103) that results from a Galerkin discretization of the parametric PDE (3.30) corresponding to the PDE (3.24) with a fuzzy diffusion coefficient (3.28). Based on ideas from the literature on stochastic PDEs, we propose two preconditioners for solving those algebraic systems. The first preconditioner is the fuzzy analogue to the well-known mean-based preconditioner [174], whereas the second one is of multigrid type and related to the algorithms in [68, 133, 186]. By means of a local Fourier analysis (LFA), we show that the convergence properties are optimal w.r.t. the discretization parameters, i.e., the size of the spatial mesh and the number of Chebyshev polynomials. Robustness against the magnitude of the input uncertainty is also investigated. Finally, the accuracy of the Galerkin approximation and the efficiency of the proposed iterative solvers is numerically demonstrated on two nontrivial problems. The results of this chapter have been published in [36].

4.1 Two preconditioners for the discrete system

The major computational cost in solving a fuzzy PDE by the Galerkin approach results from solving the high-dimensional $n_\psi n_\phi \times n_\psi n_\phi$ system (3.103). We

recall that system here, for the reader's convenience:

$$\left(\boldsymbol{G}_0 \otimes \boldsymbol{A}_0 + \sum_{k=1}^{d_\Xi} \boldsymbol{G}_k \otimes \boldsymbol{A}_k \right) \boldsymbol{u} = \boldsymbol{g} \otimes \boldsymbol{f}, \tag{4.1}$$

where $\boldsymbol{A}_0, \dots, \boldsymbol{A}_{d_\Xi} \in \mathbb{R}^{n_\phi \times n_\phi}$ and $\boldsymbol{G}_0, \dots, \boldsymbol{G}_{d_\Xi} \in \mathbb{R}^{n_\psi \times n_\psi}$.

We will assume that the polynomial approximation space used to discretize in the parameter domain Ξ is of isotropic structure containing the polynomials up to total degree p, i.e., we use $\mathbb{P}_{\mathcal{J}^{\mathrm{TD}}_{\boldsymbol{w},p}} \subset L^2_w(\Xi)$ with $\mathcal{J}^{\mathrm{TD}}_{\boldsymbol{w},p}$ as defined in (3.93) and with $\boldsymbol{w} = (1, 1, \dots)$. The basis functions $\psi_1, \dots, \psi_{n_\psi}$ for $\mathbb{P}_{\mathcal{J}^{\mathrm{TD}}_{\boldsymbol{w},p}}$ that we will use for the construction of the matrices $\boldsymbol{G}_0, \dots, \boldsymbol{G}_{d_\Xi}$ are the (normalized) multivariate Chebyshev polynomials up to total degree p reordered into a single sequence. The number $n_\psi = \#(\mathcal{J}^{\mathrm{TD}}_{\boldsymbol{w},p})$ of basis functions is equal to

$$n_\psi = \frac{(d_\Xi + p)!}{d_\Xi!\, p!}. \tag{4.2}$$

In the context of stochastic PDEs—which after a polynomial chaos discretization yield a system with a similar structure to (4.1)—a lot of research on iterative solvers for the high-dimensional algebraic systems has been done; see, for example, [182]. In this section, we adapt some of the popular solvers for stochastic Galerkin finite element systems to fuzzy Galerkin discretizations, and we analyze how the use of Chebyshev polynomials influences the convergence.

4.1.1 Center-based preconditioner

Similar to a mean-based preconditioner [174, 199], a straightforward preconditioner to (4.1) is given by

$$\boldsymbol{I}_{n_\psi} \otimes \boldsymbol{A}_0\,. \tag{4.3}$$

The concept of a mean value does not really make sense in a fuzzy context, so we shall use a different name. We call this preconditioner the center-based preconditioner, since it corresponds to the diffusion coefficient obtained by evaluating the parameters at the center of the parameter hyperrectangle. In practice when applying this preconditioner, the inversion of $\boldsymbol{A}_0$ is approximated by, e.g., one multigrid cycle. The convergence properties of this preconditioner follow from Theorem 4.1.1. They are summarized subsequently in Corollary 4.1.1.

Theorem 4.1.1. *With $\boldsymbol{G}_k$ and $\boldsymbol{A}_k$ defined in* (3.90), (3.97), (3.98), *using normalized multivariate Chebyshev polynomials on $[-1, 1]^{d_\Xi}$ of total degree p*

and with $w(\boldsymbol{\xi})$ given by (3.86)*, the eigenvalues λ of the generalized eigenvalue problem $\left(\sum_{k=0}^{d_\Xi} \boldsymbol{G}_k \otimes \boldsymbol{A}_k\right) \boldsymbol{u} = \lambda \left(\boldsymbol{G}_0 \otimes \boldsymbol{A}_0\right) \boldsymbol{u}$ lie in the interval $[1-\tau, 1+\tau]$, where*

$$\tau = \sigma \left\| \frac{1}{a_0} \right\|_{L^\infty(\Omega)} \sum_{k=1}^{d_\Xi} \|a_k\|_{L^\infty(\Omega)} \,. \tag{4.4}$$

Proof. The eigenvalues λ can be written as $\lambda = \theta + 1$, where θ satisfies

$$\sum_{k=1}^{d_\Xi} \left(\boldsymbol{G}_0 \otimes \boldsymbol{A}_0\right)^{-1} \left(\boldsymbol{G}_k \otimes \boldsymbol{A}_k\right) \boldsymbol{v} = \theta \boldsymbol{v} \,.$$

Using properties of the Kronecker product and the fact that $\boldsymbol{G}_0$ is the identity matrix results in

$$\sum_{k=1}^{d_\Xi} \left(\boldsymbol{G}_k \otimes \boldsymbol{A}_0^{-1} \boldsymbol{A}_k\right) \boldsymbol{v} = \theta \boldsymbol{v} \,. \tag{4.5}$$

Applying Lemma 3.2 in [174], we find that the eigenvalues of $\boldsymbol{A}_0^{-1} \boldsymbol{A}_k$ belong to the interval

$$[-\gamma, \gamma] \qquad \text{with} \quad \gamma = \sigma \left\| \frac{1}{a_0} \right\|_{L^\infty(\Omega)} \|a_k\|_{L^\infty(\Omega)} \,. \tag{4.6}$$

From [186], we have that the eigenvalues of $\boldsymbol{G}_k$ lie in the interval $[\zeta_{p,1}, \zeta_{p,p}]$, where $\zeta_{p,1}$ is the smallest zero and $\zeta_{p,p}$ the largest zero of a univariate Chebyshev polynomial of degree p. Since the zeros of a Chebyshev polynomial on $[-1,1]$ are bounded by -1 and 1, the eigenvalues of the Kronecker product $\boldsymbol{G}_k \otimes \boldsymbol{A}_0^{-1} \boldsymbol{A}_k$ lie also in the interval given by (4.6). Bounding the eigenvalues of (4.5) by the eigenvalue bounds of the matrices in the sum yields

$$-\tau \leq \theta_{\min} \qquad \text{and} \qquad \theta_{\max} \leq \tau$$

with τ specified as (4.4). From this, the result follows. □

Corollary 4.1.1. *The number of iterations required to solve the algebraic system* (4.1) *with the conjugate gradients (CG) method, preconditioned by the center-based preconditioner* (4.3)*, is independent of the mesh size h and of the polynomial degree p when a Chebyshev polynomial discretization is used to define the matrices $\boldsymbol{G}_k$ in* (3.90).

From Theorem 4.1.1, we do notice that the convergence may degrade when large parameter variations occur (e.g., by a large value of σ).

4.1.2 Collective smoothing multigrid method

Based on earlier positive experiences with multigrid for stochastic Galerkin finite element discretizations [68, 133, 181, 186], we also propose a multigrid strategy for the fuzzy Galerkin systems (4.1). Basically, multigrid is an iterative method which combines a smoothing operation $\boldsymbol{S}_l$ and a coarse-grid correction in a recursive manner; see Algorithm 1. A prolongation operator $\boldsymbol{P}_{l-1,l}$ and a restriction operator $\boldsymbol{R}_{l,l-1}$ transfer corrections and residuals back and forth between different levels $l = l_{\max}, l_{\max}-1, \ldots, 0$, where $l = l_{\max}$ represents the finest level and $l = 0$ represents the coarsest level. On each level l, there is a corresponding operator $\boldsymbol{L}_l$.

Algorithm 1 Multigrid iteration

function MULTIGRID(approximation $\boldsymbol{u}_l$, right hand side $\boldsymbol{f}_l$, level l)
  if $l = 0$ **then**
    $\boldsymbol{u}_0 \leftarrow \boldsymbol{L}_0^{-1} \boldsymbol{f}_0$
  else
    Pre-smoothing: apply smoother ν_1 times, i.e., $\boldsymbol{u}_l \leftarrow \boldsymbol{S}_l^{\nu_1}(\boldsymbol{u}_l, \boldsymbol{L}_l, \boldsymbol{f}_l)$
    Restrict residual to coarser level: $\boldsymbol{f}_{l-1} \leftarrow \boldsymbol{R}_{l,l-1}(\boldsymbol{f}_l - \boldsymbol{L}_l \boldsymbol{u}_l)$
    Coarse grid correction:
    $\boldsymbol{e}_{l-1} \leftarrow 0$
    for $i = 1, \ldots, \gamma$ **do**
      $\boldsymbol{e}_{l-1} \leftarrow$ MULTIGRID($\boldsymbol{e}_{l-1}, \boldsymbol{f}_{l-1}, l-1$)
    end for
    Prolongate correction and update approximation:

$$\boldsymbol{u}_l \leftarrow \boldsymbol{u}_l + \boldsymbol{P}_{l-1,l} \boldsymbol{e}_{l-1}$$

    Post-smoothing: apply smoother ν_2 times, i.e.,

$$\boldsymbol{u}_l \leftarrow \boldsymbol{S}_l^{\nu_2}(\boldsymbol{u}_l, \boldsymbol{L}_l, \boldsymbol{f}_l)$$

  end if
  return $\boldsymbol{u}_l$
end function

The intergrid transfer operators $\boldsymbol{P}_{l-1,l}$ and $\boldsymbol{R}_{l,l-1}$ and the operators $\boldsymbol{L}_l$ are constructed during setup of the multigrid method. In algebraic multigrid, for example, the operators $\boldsymbol{P}_{l-1,l}$ and $\boldsymbol{R}_{l,l-1}$ are constructed from a repeated algebraic coarsening of the system matrix. The operators $\boldsymbol{L}_l$ are constructed based on the Galerkin principle:

$$\boldsymbol{L}_{l-1} = \boldsymbol{P}_{l-1,l} \boldsymbol{L}_l \boldsymbol{R}_{l,l-1} \tag{4.7}$$

with $\boldsymbol{L}_{l_{\max}}$ equal to the system matrix.

Smoothing operator

We suggest the use of a collective smoothing operator. Two variants will be considered. In the block Gauss–Seidel smoother each mesh point is visited sequentially, one after the other, and, locally, a linear system of size $n_\psi \times n_\psi$ is solved. This linear system couples all degrees of freedom which are physically located at that particular grid point. The second variant is the block Jacobi smoother, where the values at the different grid points are updated in parallel. In terms of a classical matrix splitting iteration, the smoothing operator can be written as

$$\left(\sum_{k=0}^{d_\Xi} \boldsymbol{G}_k \otimes \boldsymbol{A}_k^+ \right) \boldsymbol{u}^{\text{new}} = \boldsymbol{g} \otimes \boldsymbol{f} - \left(\sum_{k=0}^{d_\Xi} \boldsymbol{G}_k \otimes \boldsymbol{A}_k^- \right) \boldsymbol{u}^{\text{old}} \tag{4.8}$$

with $\boldsymbol{A}_k^- = \boldsymbol{A}_k - \boldsymbol{A}_k^+$. Matrix $\boldsymbol{A}_k^+$ equals the (scaled) diagonal part of $\boldsymbol{A}_k$ or the lower triangular part in the case of a block Jacobi or a block Gauss–Seidel smoother, respectively. Every smoothing iteration (4.8) entails the solution of n_ϕ sparse systems, each of size $n_\psi \times n_\psi$. These systems can be solved, for example, by a direct solver.

Intergrid transfer operators and coarse grid operators

Coarsening will be done in the spatial dimensions only. A hierarchy of spatial grids is constructed, and the same hierarchy is used for each of the fuzzy unknowns. For the Fourier analysis and accompanying numerical results in §4.2, we use standard coarsening by a factor of 2 in each spatial dimension, and the coarse grid operator is constructed by rediscretization. For the irregular mesh numerical results in §4.3, an algebraic coarsening is used based on the matrix $\boldsymbol{A}_0$. The latter matrix is provided as input to a standard algebraic multigrid (AMG) code for deterministic PDEs, which analyzes the properties of the matrix and generates a sequence of meshes together with the corresponding intergrid transfer operators. Denote by $\widehat{\boldsymbol{R}}_{l,l-1}$ a restriction operator originating from a multigrid hierarchy for the deterministic matrix $\boldsymbol{A}_0$ at multigrid level l, and by $\widehat{\boldsymbol{P}}_{l-1,l}$, the corresponding prolongation operator. The intergrid transfer operators for the system (4.1) are then given by

$$\boldsymbol{R}_{l,l-1} = \boldsymbol{I}_{n_\psi} \otimes \widehat{\boldsymbol{R}}_{l,l-1} \qquad \text{and} \qquad \boldsymbol{P}_{l-1,l} = \boldsymbol{I}_{n_\psi} \otimes \widehat{\boldsymbol{P}}_{l-1,l}\,. \tag{4.9}$$

The coarse grid operators $\boldsymbol{L}_l$ are constructed according to the Galerkin principle (4.7) with

$$\boldsymbol{L}_{l_{\max}} = \left(\boldsymbol{G}_0 \otimes \boldsymbol{A}_0 + \sum_{k=1}^{d_\Xi} \boldsymbol{G}_k \otimes \boldsymbol{A}_k \right). \tag{4.10}$$

4.2 Local Fourier mode convergence analysis

In [182, 186], an LFA of multigrid applied to a stochastic Galerkin discretization of stochastic elliptic problems is detailed and numerical results are given for the case of a Hermite or Legendre polynomial stochastic discretization. The LFA of multigrid applied to fuzzy PDEs proceeds similarly, but the modified definition of the fuzzy discretization matrices has to be taken into account, which now uses Chebyshev polynomials. Below we summarize the main components of the Fourier analysis. Our model problem and its discretization are such that a direct comparison is possible with the corresponding results for the stochastic Galerkin case in [182, 186]. The considered problem is a two-dimensional diffusion equation (4.11), discretized by finite differences. In principle, a Fourier analysis for a spatial finite element discretization could also be possible. However, the latter is somewhat technically more involved, especially for variable coefficient problems, and does not really lead to an additional insight for the problem considered here. The numerical results in §4.2.5 show the effect of the Chebyshev polynomial discretization on the convergence factors.

4.2.1 Model problem for LFA

We apply an LFA analysis to the following two-dimensional model problem:

$$-\frac{\partial^2 u(\boldsymbol{x},\widetilde{\boldsymbol{\xi}})}{\partial x_1^2} - a(\boldsymbol{x},\widetilde{\boldsymbol{\xi}})\frac{\partial^2 u(\boldsymbol{x},\widetilde{\boldsymbol{\xi}})}{\partial x_2^2} = f(\boldsymbol{x}) \qquad \text{in } \Omega\,, \tag{4.11}$$

where $\widetilde{\boldsymbol{\xi}} := (\widetilde{\xi}_1, \ldots, \widetilde{\xi}_{d_\Xi}) \in \mathcal{F}(\mathbb{R}^{d_\Xi})$ is a vector of noninteractive fuzzy numbers and $\Xi := [\widetilde{\boldsymbol{\xi}}]_0$ is the parameter domain. An infinite spatial domain $\Omega := \mathbb{R}^{d_\Omega}$ is assumed in order to eliminate the effect of boundary conditions. We apply the fuzzy Galerkin discretization described in §3.3.5 to (4.11). The spatial discretization uses a standard five-point finite difference scheme on a rectangular grid $\Omega_h = \{(ih, jh)\}_{i,j\in\mathbb{Z}}$ with grid spacing h in x_1- and x_2-directions. Using the orthonormality of the Chebyshev polynomials, we arrive

at the following algebraic system:

$$-\left(\boldsymbol{u}_{i-1,j}-2\boldsymbol{u}_{i,j}+\boldsymbol{u}_{i+1,j}\right)-\boldsymbol{M}_{i,j}\left(\boldsymbol{u}_{i,j-1}-2\boldsymbol{u}_{i,j}+\boldsymbol{u}_{i,j+1}\right) = h^2 f_{i,j}\boldsymbol{g}\,, \quad (4.12)$$

where $\boldsymbol{u}_{i,j} \in \mathbb{R}^{n_\psi \times 1}$ represents the discrete approximation to u at grid point (ih, jh), $f_{i,j}$ corresponds to $f(ih, jh)$, and $\boldsymbol{g}$ is defined as in (3.91). The fuzzy discretization is captured by the $(n_\psi \times n_\psi)$-matrix $\boldsymbol{M}_{i,j}$, which is defined as

$$\boldsymbol{M}_{i,j|r,s}(\boldsymbol{x}) = \int_\Xi a(\boldsymbol{x},\boldsymbol{\xi})\psi_r(\boldsymbol{\xi})\psi_s(\boldsymbol{\xi})w(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}\,.$$

Note that in the case of a variable coefficient problem, LFA is performed by freezing each coefficient to its value at the considered grid point (ih, jh) [213]. For problem (4.11), this corresponds to replacing the fuzzy field $a(\boldsymbol{x}, \widetilde{\boldsymbol{\xi}})$ by a fuzzy number $a(\widetilde{\boldsymbol{\xi}})$. As such, $\boldsymbol{M}_{i,j}$ can be considered to be a fixed known matrix, which will be denoted as $\boldsymbol{M}$.

When the equations are collected over all grid points, a linear system of equations results,

$$\boldsymbol{L}_h\boldsymbol{u}_h = \boldsymbol{f}_h\,. \quad (4.13)$$

The dimension of $\boldsymbol{L}_h$ equals the number of spatial grid points multiplied by n_ψ.

4.2.2 Local Fourier representation

In order to set up an LFA, we decompose the iteration error into a sum of exponential Fourier grid modes of the form

$$\boldsymbol{e}(\boldsymbol{\theta},\boldsymbol{z}) = \exp\left(\imath\left(i\theta_{x_1}+j\theta_{x_2}\right)\right)\boldsymbol{z}\,, \quad (4.14)$$

where $\boldsymbol{z} \in \mathbb{R}^{n_\psi}$, $\boldsymbol{\theta} := (\theta_{x_1}, \theta_{x_2}) \in [-\pi, \pi)^2$, and $\imath$ represents the imaginary unit. Note that the linear discrete operator $\boldsymbol{L}_h$ in (4.13) is invariant to $\boldsymbol{e}(\boldsymbol{\theta}, \boldsymbol{z})$:

$$\boldsymbol{L}_h\boldsymbol{e}(\boldsymbol{\theta},\boldsymbol{z}) = \widehat{\boldsymbol{L}}_h(\boldsymbol{\theta})\boldsymbol{e}(\boldsymbol{\theta},\boldsymbol{z})\,, \quad (4.15)$$

where the symbol $\widehat{\boldsymbol{L}}_h(\boldsymbol{\theta})$ of $\boldsymbol{L}_h$ is defined as

$$\begin{aligned}\widehat{\boldsymbol{L}}_h(\boldsymbol{\theta}) &= \frac{1}{h^2}\Big(\boldsymbol{I}_{n_\psi}\left(\exp(-\imath\theta_{x_1})-2+\exp(\imath\theta_{x_1})\right)\\ &\qquad +\boldsymbol{M}\left(\exp(-\imath\theta_{x_2})-2+\exp(\imath\theta_{x_2})\right)\Big)\\ &= -\frac{4}{h^2}\left(\sin^2\left(\theta_{x_1}/2\right)\boldsymbol{I}_{n_\psi}+\sin^2\left(\theta_{x_2}/2\right)\boldsymbol{M}\right)\,.\end{aligned}$$

If $\boldsymbol{z}$ in (4.14) is selected to be one of the eigenvectors $\boldsymbol{z}_m$ of $\boldsymbol{M}$ with corresponding eigenvalue λ_m, equality (4.15) simplifies to

$$\boldsymbol{L}_h \boldsymbol{e}(\boldsymbol{\theta}, \boldsymbol{z}_m) = \widehat{\boldsymbol{L}}_h(\boldsymbol{\theta}, \lambda_m) \boldsymbol{e}(\boldsymbol{\theta}, \boldsymbol{z}_m) \tag{4.16}$$

with

$$\widehat{\boldsymbol{L}}_h(\boldsymbol{\theta}, \lambda_m) = -\frac{4}{h^2} \left(\sin^2 \left(\theta_{x_1}/2 \right) + \sin^2 \left(\theta_{x_2}/2 \right) \lambda_m \right) .$$

Hence, the Fourier mode $\boldsymbol{e}(\boldsymbol{\theta}, \boldsymbol{z}_m)$ is an eigenfunction of the (frozen) discrete differential operator.

4.2.3 Smoothing analysis

For many stationary iterative methods, e.g., Jacobi and lexicographic Gauss–Seidel iterations, the Fourier modes (4.14) are eigenfunctions of the corresponding iteration operator $\boldsymbol{S}_h$. The corresponding eigenvalues are called the amplification factor or Fourier symbol of the iteration operator, denoted by $\widehat{\boldsymbol{S}}_h(\boldsymbol{\theta})$, and determine the asymptotic convergence factor. For a variable coefficient problem with a sufficiently smooth coefficient a, this convergence factor is defined as

$$\rho = \max_{\boldsymbol{x}=(ih,jh)\in\Omega_h} \max_{\boldsymbol{\theta}\in[-\pi,\pi)^2} \rho \left(\widehat{\boldsymbol{S}}_h(\boldsymbol{\theta}) \right) \tag{4.17}$$

with $\rho \left(\widehat{\boldsymbol{S}}_h(\boldsymbol{\theta}) \right)$ the spectral radius of $\widehat{\boldsymbol{S}}_h(\boldsymbol{\theta})$.

Applying the block Gauss–Seidel and damped block Jacobi iteration operator characterized by (4.8) to the Fourier mode (4.14), we find that

$$\boldsymbol{S}_h \boldsymbol{e}(\boldsymbol{\theta}, \boldsymbol{z}_m) = \widehat{\boldsymbol{S}}_h(\boldsymbol{\theta}, \lambda_m) \boldsymbol{e}(\boldsymbol{\theta}, \boldsymbol{z}_m),$$

where the symbols $\widehat{\boldsymbol{S}}_h$ are, respectively, given by

$$\widehat{\boldsymbol{S}}_h^{\mathrm{GS}}(\boldsymbol{\theta}, \lambda_m) = \frac{\exp(\imath\theta_{x_1}) + \exp(\imath\theta_{x_2})\lambda_m}{\left(2 - \exp(-\imath\theta_{x_1})\right) + \left(2 - \exp(-\imath\theta_{x_2})\right) \lambda_m} ,$$

$$\widehat{\boldsymbol{S}}_h^{\mathrm{JAC}}(\boldsymbol{\theta}, \lambda_m) = 1 - \omega + \omega \frac{\cos(\theta_{x_1}) + \cos(\theta_{x_2})\lambda_m}{1 + \lambda_m} .$$

Here, ω represents the Jacobi damping factor and λ_m equals the eigenvalue of $\boldsymbol{M}$ corresponding to the eigenvector $\boldsymbol{z}_m$. Proceeding analogously to [182, 186], the optimal damping factor ω can be determined, as can the symbols of other classical (block) splitting iterations.

4.2.4 Two-grid analysis

To determine the action of a two-grid operator on the Fourier modes (4.14), the Fourier space is decomposed into subspaces spanned by four harmonics, $\mathcal{H}(\boldsymbol{\theta}, \boldsymbol{z}) := \text{span}[\boldsymbol{e}(\boldsymbol{\theta}_1, \boldsymbol{z}), \boldsymbol{e}(\boldsymbol{\theta}_2, \boldsymbol{z}), \boldsymbol{e}(\boldsymbol{\theta}_3, \boldsymbol{z}), \boldsymbol{e}(\boldsymbol{\theta}_4, \boldsymbol{z})]$, for a given $(\theta_{x_1}, \theta_{x_2}) \in [-\frac{\pi}{2}, \frac{\pi}{2})^2$ with

$$\boldsymbol{\theta}_1 = (\theta_{x_1}, \theta_{x_2}), \qquad \boldsymbol{\theta}_3 = (\theta_{x_1} - \text{sign}(\theta_{x_1})\pi, \theta_{x_2}),$$

$$\boldsymbol{\theta}_2 = (\theta_{x_1}, \theta_{x_2} - \text{sign}(\theta_{x_2})\pi), \qquad \boldsymbol{\theta}_4 = (\theta_{x_1} - \text{sign}(\theta_{x_1})\pi, \theta_{x_2} - \text{sign}(\theta_{x_2})\pi).$$

These spaces are invariant under the fine and coarse grid discrete differential operators and under certain smoothing operators. The action of a smoothing operator on an element of such a space can be described by a (4×4) diagonal matrix $\widehat{\mathcal{S}}_h(\boldsymbol{\theta}, \lambda_m)$ with

$$\widehat{\mathcal{S}}_h(\boldsymbol{\theta}, \lambda_m) := \text{diag}(\widehat{\boldsymbol{S}}_h(\boldsymbol{\theta}_1, \lambda_m),\ \widehat{\boldsymbol{S}}_h(\boldsymbol{\theta}_2, \lambda_m),\ \widehat{\boldsymbol{S}}_h(\boldsymbol{\theta}_3, \lambda_m),\ \widehat{\boldsymbol{S}}_h(\boldsymbol{\theta}_4, \lambda_m))\,.$$

A similar diagnonal matrix representation for the action of $\boldsymbol{L}_h$ holds and is denoted by $\widehat{\mathcal{L}}_h$. On the coarse grid, $\boldsymbol{L}_{2h}$ is constructed by discretizing (4.11) with a standard five-point finite difference scheme on a rectangular grid with grid spacing $2h$. Its action can be represented by

$$\widehat{\boldsymbol{L}}_{2h}(\boldsymbol{\theta}, \lambda_m) := -\frac{1}{h^2}\left(\sin^2(\theta_{x_1}) + \sin^2(\theta_{x_2})\lambda_m\right).$$

The prolongation operator (4.9) maps the mode $\boldsymbol{e}(2\boldsymbol{\theta}, \boldsymbol{z})$ onto $\mathcal{H}(\boldsymbol{\theta}, \boldsymbol{z})$ [197]. In the case of bilinear interpolation, it is characterized by the symbol $\widehat{\boldsymbol{P}}_{2h,h}(\boldsymbol{\theta})$, which is given by

$$\widehat{\boldsymbol{P}}_{2h,h}(\boldsymbol{\theta}) := \frac{1}{4}\begin{bmatrix} (1+\cos(\theta_{x_1}))(1+\cos(\theta_{x_2})) \\ (1+\cos(\theta_{x_1}))(1-\cos(\theta_{x_2})) \\ (1-\cos(\theta_{x_1}))(1+\cos(\theta_{x_2})) \\ (1-\cos(\theta_{x_1}))(1-\cos(\theta_{x_2})) \end{bmatrix}. \tag{4.18}$$

Using standard coarsening, the restriction operator maps the space $\mathcal{H}(\boldsymbol{\theta}, \boldsymbol{z})$ onto the single mode $\boldsymbol{e}(\boldsymbol{\theta}_1, \boldsymbol{z})$. The corresponding Fourier representation is given by $\widehat{\boldsymbol{R}}_{h,2h}(\boldsymbol{\theta}) = (\widehat{\boldsymbol{P}}_{2h,h}(\boldsymbol{\theta}))^T$.

In summary, the action of the two-grid operator, specified in §4.1.2 and applied to the differential operator (4.12) on the space $\mathcal{H}(\boldsymbol{\theta}, \boldsymbol{z})$, is characterized by

$$\begin{aligned}\widehat{\boldsymbol{T}}_h(\boldsymbol{\theta}, \lambda_m) := {} & (\widehat{\mathcal{S}}_h(\boldsymbol{\theta}, \lambda_m))^{\nu_2} \\ & \cdot \left(\boldsymbol{I}_4 - \widehat{\boldsymbol{P}}_{2h,h}(\boldsymbol{\theta})\left(\widehat{\boldsymbol{L}}_{2h}(\boldsymbol{\theta}, \lambda_m)\right)^{-1}\widehat{\boldsymbol{R}}_{h,2h}(\boldsymbol{\theta})\widehat{\mathcal{L}}_h(\boldsymbol{\theta}, \lambda_m)\right) \\ & \cdot (\widehat{\mathcal{S}}_h(\boldsymbol{\theta}, \lambda_m))^{\nu_1}\,,\end{aligned} \tag{4.19}$$

where ν_1 and ν_2 are the number of presmoothing, respectively, postsmoothing, steps, and $\boldsymbol{I}_4 \in \boldsymbol{R}^{4\times 4}$ is an identity matrix. Under the assumption that the variation of the coefficient a is sufficiently smooth, the asymptotic convergence factor of the two-grid scheme is defined as

$$\rho_{TG} = \max_{\boldsymbol{x}=(ih,jh)\in\Omega_h} \max_{\lambda_m\in\sigma(\boldsymbol{M}_{i,j})} \max_{\boldsymbol{\theta}\in[-\frac{\pi}{2},\frac{\pi}{2})^2} \rho\left(\widehat{\boldsymbol{T}}_h(\boldsymbol{\theta},\lambda_m)\right), \tag{4.20}$$

where $\sigma(\boldsymbol{M}_{i,j})$ represents the spectrum of $\boldsymbol{M}_{i,j}$.

4.2.5 Numerical results

We shall demonstrate the correctness and accuracy of the Fourier analysis and comment on the convergence properties of the methods proposed in §4.1. We consider model problem (4.11) on a unit square $\Omega = [0,1]^2$ with zero Dirichlet boundary conditions. The fuzzy diffusion coefficient $\widetilde{a}$ is given by a linear combination of d_Ξ triangular fuzzy numbers as in (3.28). The functions a_k are constructed as $a_k = \sqrt{\kappa_k} v_k$ with κ_k and v_k, respectively, the eigenvalues and eigenfunctions of an exponential kernel $C(\boldsymbol{x}_1,\boldsymbol{x}_2) = \exp(-\|\boldsymbol{x}_1-\boldsymbol{x}_2\|_1 / L_c)$ with $L_c = 1$. See Fig. 4.1 for a plot of some of the a_k.

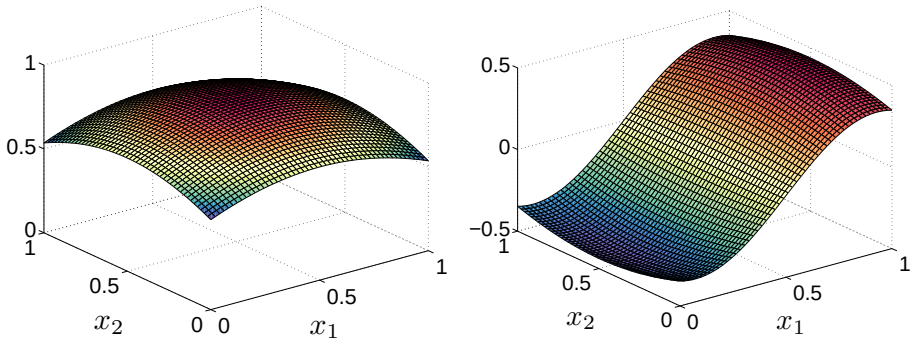
Collective smoothing multigrid method

Table 4.1 shows the theoretical multigrid convergence factors obtained by the LFA for various choices of the discretization parameters. These values were obtained numerically from (4.20) by an exhaustive search over the grid Ω_h, the spectrum $\sigma(\boldsymbol{M}_{i,j})$ and a fine grid sampling of $[-\frac{\pi}{2},\frac{\pi}{2})^2$. The spectrum $\sigma(\boldsymbol{M}_{i,j})$ was computed numerically from the explicitly constructed matrices $\boldsymbol{M}_{i,j}$. Further, the table also provides numerically observed convergence factors, obtained with an implementation of the algorithm. The numerical results confirm the accuracy of the LFA results.

Table 4.1 demonstrates the robust convergence behavior of multigrid: the convergence factors are independent of the spatial and fuzzy discretization parameters and are only slightly influenced by the width of the support of the fuzzy field, as determined by σ.

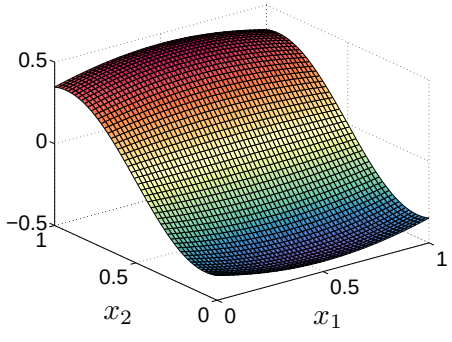
Center-based preconditioner

The LFA presented for the collective smoothing multigrid method can also be applied to other iterative methods. As a standalone solver, the center-

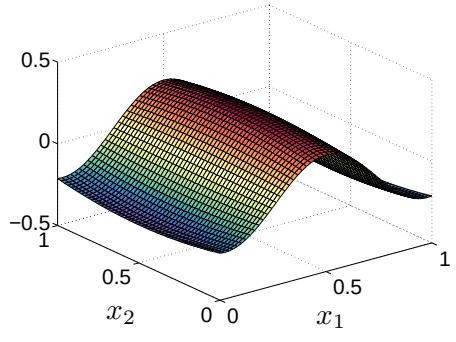


(a) a_1

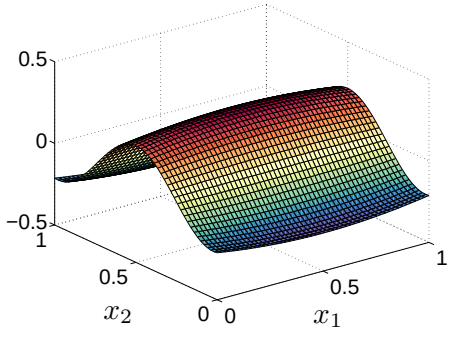
(b) a_2



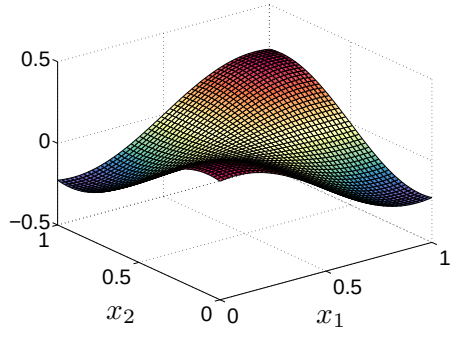
(c) a_3



(d) a_4



(e) a_5



(f) a_6

Figure 4.1: Plots of $a_k = \sqrt{\kappa_k} v_k$ with κ_k and v_k, respectively, the eigenvalues and eigenfunctions of an exponential kernel $C(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp(-\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_1 / L_c)$ with $L_c = 1$.

Table 4.1: Numerical (ρ_{num}) and theoretical (ρ_{theo}) convergence factors for the two-grid cycle TG(2,1) (i.e., $l_{\max} = 1$, $\gamma = 1$, $\nu_1 = 2$, $\nu_2 = 1$ in Algorithm 1) with lexicographic Gauss–Seidel smoother. The default parameters are $h = 2^{-5}$, $d_\Xi = 4$, $p = 2$, $\sigma/a_0 = 0.2$. Every block row corresponds to the case when one of those default values is varied.

grid spacing h	$h = 2^{-4}$	$h = 2^{-5}$	$h = 2^{-6}$	$h = 2^{-7}$
ρ_{theo}	0.120	0.120	0.120	0.120
ρ_{num}	0.106	0.110	0.112	0.112
fuzzy numbers d_Ξ	$d_\Xi = 1$	$d_\Xi = 5$	$d_\Xi = 8$	$d_\Xi = 10$
ρ_{theo}	0.119	0.120	0.120	0.120
ρ_{num}	0.109	0.111	0.111	0.110
polynomial degree p	$p = 1$	$p = 3$	$p = 4$	$p = 5$
ρ_{theo}	0.119	0.120	0.121	0.122
ρ_{num}	0.111	0.111	0.110	0.111
scaling factor σ/a_0	$\sigma/a_0 = 0.1$	$\sigma/a_0 = 0.4$	$\sigma/a_0 = 0.6$	$\sigma/a_0 = 0.7$
ρ_{theo}	0.119	0.127	0.175	0.247
ρ_{num}	0.109	0.114	0.160	0.221

based preconditioner can be interpreted as block Jacobi method. For the model problem, it can be formulated as

$$
\begin{aligned}
&- \left(\boldsymbol{u}^{\text{new}}_{i-1,j} - 2\boldsymbol{u}^{\text{new}}_{i,j} + \boldsymbol{u}^{\text{new}}_{i+1,j}\right) - \boldsymbol{M}^{+}\left(\boldsymbol{u}^{\text{new}}_{i,j-1} - 2\boldsymbol{u}^{\text{new}}_{i,j} + \boldsymbol{u}^{\text{new}}_{i,j+1}\right) \\
&\qquad = h^2 f_{i,j}\boldsymbol{g} + \boldsymbol{M}^{-}\left(\boldsymbol{u}^{\text{old}}_{i,j-1} - 2\boldsymbol{u}^{\text{old}}_{i,j} + \boldsymbol{u}^{\text{old}}_{i,j+1}\right), \qquad i,j = 1,\ldots, \quad (4.21)
\end{aligned}
$$

where $\boldsymbol{M}^{+}$ equals the diagonal part of $\boldsymbol{M}$ and $\boldsymbol{M}^{-} = \boldsymbol{M} - \boldsymbol{M}^{+}$. The equivalence between the preconditioner (4.3) and the iteration (4.21) follows from the orthonormality of the polynomial basis. Indeed, $\boldsymbol{G}_0 \equiv \boldsymbol{I}_{n_\psi}$ and the diagonal elements of $\boldsymbol{G}_k$, $k = 1,\ldots,d_\Xi$ are zero, as proved in [182].

Table 4.2 illustrates the LFA convergence factors of (4.21). The computation of these convergence factors is based on the LFA derivation given in [182], adapted to the use of Chebyshev polynomials. Although the convergence factors in Table 4.2 behave less regularly than in Table 4.1, we observe an asymptotic independence of the convergence behavior on the discretization parameters. This property is also confirmed by Theorem 4.1.1.

As a straightforward extension to iteration (4.21), we can also consider the block Gauss–Seidel variant, in which case $\boldsymbol{M}^{+}$ in (4.21) equals the lower triangular part of $\boldsymbol{M}$. The LFA convergence factors of the block Gauss–Seidel case are given in Table 4.3. Comparing the results in Table 4.1 to those in Table 4.3, we note the smaller convergence factors of the block Gauss–Seidel method in

Table 4.2: Theoretical (ρ_{theo}) and numerical (ρ_{num}) convergence factors for the block Jacobi iteration (4.21) applied to the LFA model problem (4.11). (Default: $h = 2^{-5}$, $d_\Xi = 4$, $p = 2$, $\sigma/a_0 = 0.2$)

grid spacing h	$h = 2^{-4}$	$h = 2^{-5}$	$h = 2^{-6}$	$h = 2^{-7}$
ρ_{theo}	0.197	0.199	0.199	0.199
ρ_{num}	0.184	0.189	0.192	0.192
fuzzy numbers d_Ξ	$d_\Xi = 1$	$d_\Xi = 5$	$d_\Xi = 8$	$d_\Xi = 10$
ρ_{theo}	0.147	0.202	0.213	0.214
ρ_{num}	0.140	0.194	0.205	0.207
polynomial degree p	$p = 1$	$p = 3$	$p = 4$	$p = 5$
ρ_{theo}	0.130	0.242	0.270	0.287
ρ_{num}	0.125	0.227	0.253	0.270
scaling factor σ/a_0	$\sigma/a_0 = 0.1$	$\sigma/a_0 = 0.4$	$\sigma/a_0 = 0.6$	$\sigma/a_0 = 0.7$
ρ_{theo}	0.0993	0.397	0.596	0.695
ρ_{num}	0.0947	0.380	0.575	0.672

Table 4.3: Theoretical (ρ_{theo}) and numerical (ρ_{num}) convergence factors for the block Gauss–Seidel iteration (4.21) applied to the LFA model problem (4.11). (Default: $h = 2^{-5}$, $d_\Xi = 4$, $p = 2$, $\sigma/a_0 = 0.2$)

grid spacing h	$h = 2^{-4}$	$h = 2^{-5}$	$h = 2^{-6}$	$h = 2^{-7}$
ρ_{theo}	0.0388	0.0395	0.0396	0.0397
ρ_{num}	0.0350	0.0354	0.361	0.362
fuzzy numbers d_Ξ	$d_\Xi = 1$	$d_\Xi = 5$	$d_\Xi = 8$	$d_\Xi = 10$
ρ_{theo}	0.0216	0.0406	0.0455	0.0459
ρ_{num}	0.0275	0.0374	0.0442	0.0447
polynomial degree p	$p = 1$	$p = 3$	$p = 4$	$p = 5$
ρ_{theo}	0.0168	0.0587	0.0727	0.0824
ρ_{num}	0.0160	0.0540	0.0657	0.0762
scaling factor σ/a_0	$\sigma/a_0 = 0.1$	$\sigma/a_0 = 0.4$	$\sigma/a_0 = 0.6$	$\sigma/a_0 = 0.7$
ρ_{theo}	0.00987	0.158	0.355	0.483
ρ_{num}	0.00862	0.144	0.331	0.453

comparison to the multigrid method. Each iteration of (4.21) is, however, substantially more expensive than one multigrid iteration since several systems of the size of the number of deterministic unknowns have to be solved during every iteration. The same holds for the method corresponding to Table 4.2. These expensive solves will be approximated by one deterministic multigrid cycle when applying the center-based preconditioner to practical examples.

4.3 Numerical experiments

In this section, we verify the accuracy of our spectral Galerkin discretization approach numerically, and we demonstrate the convergence properties of the proposed iterative solvers on two model problems. The accuracy of the computed solutions will be assessed in the $C(\Xi; H_0^1(\Omega))$-norm; see Corollary 3.2.1. In all numerical experiments, we iterate until the Euclidean norm of the relative residual is smaller than 10^{-9}.

4.3.1 Diffusion equation on an L-shaped domain

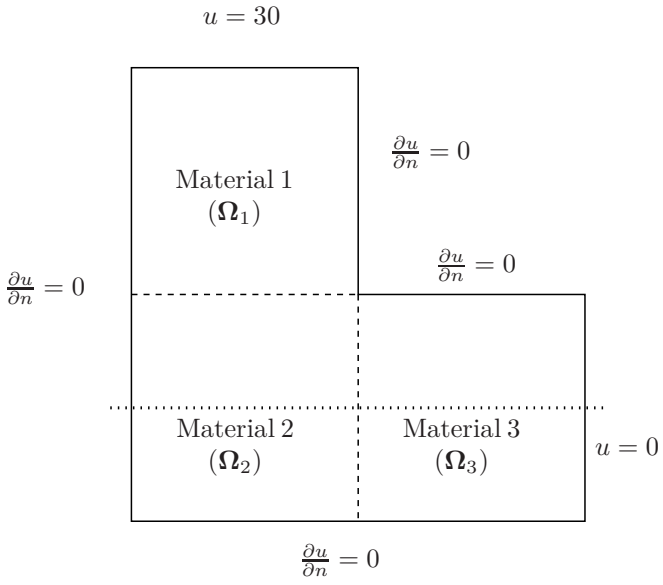
Problem setup

We consider the diffusion equation (3.24) on an L-shaped domain, as depicted in Fig. 4.2a, together with Dirichlet and Neumann boundary conditions. The source term f is set to zero. The spatial domain is partitioned in three regions, $\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3$. The fuzzy diffusion coefficient is modeled independently in each of the subdomains by an expansion of the form (3.28) with, respectively, $d_{\Xi,1}$, $d_{\Xi,2}$, and $d_{\Xi,3}$ terms. The total number of fuzzy parameters in the model is then $d_\Xi = d_{\Xi,1} + d_{\Xi,2} + d_{\Xi,3}$. For each of the subdomains, we model the $\widetilde{\xi}_k$ as noninteractive symmetrical triangular fuzzy numbers with support $[-1, 1]$. The functions a_k are constructed as $\sqrt{\kappa_k} v_k$ with κ_k and v_k the eigenvalues and eigenfunctions of an exponential kernel $C(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp(-\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_1 / L_c)$ with $L_c = 0.7$ on Ω_1, $L_c = 0.3$ on Ω_2, and $L_c = 0.1$ on Ω_3. The functions a_0 in (3.28) are taken equal to 30, 5, and 100 on subdomains Ω_1, Ω_2, and Ω_3, respectively. Finally, we apply $\sigma = 10$ on Ω_1, $\sigma = 2$ on Ω_2, and $\sigma = 20$ on Ω_3.

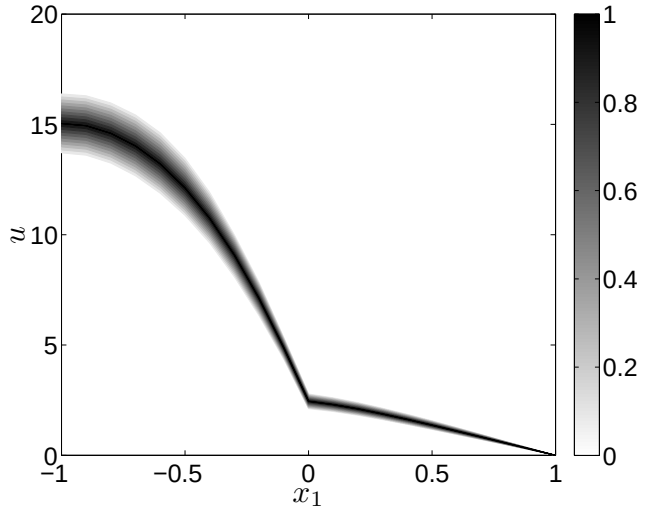
The fuzzy solution is depicted in Fig. 4.2b and Fig. 4.3. A Chebyshev discretization of degree $p = 3$ of the fuzzy parameter domain with $d_{\Xi,1} = 1$, $d_{\Xi,2} = 2$, and $d_{\Xi,3} = 3$ is applied, along with a spatial triangular finite element mesh using $n_\phi = 32\,499$ degrees of freedom.

Accuracy of spectral Galerkin approximation

Fig. 4.4a and Fig. 4.5 illustrate the accuracy of the Chebyshev response surface obtained via the spectral Galerkin approach for different sets of parameters. Only the error by the discretization in the fuzzy dimension is considered, not the error from the spatial discretization. The size of the fuzzy error can be estimated by using Theorem 3.2.1, which provides an upper bound for the distance between two fuzzy fields. That is, we compute the $C(\Xi; H_0^1(\Omega))$-norm

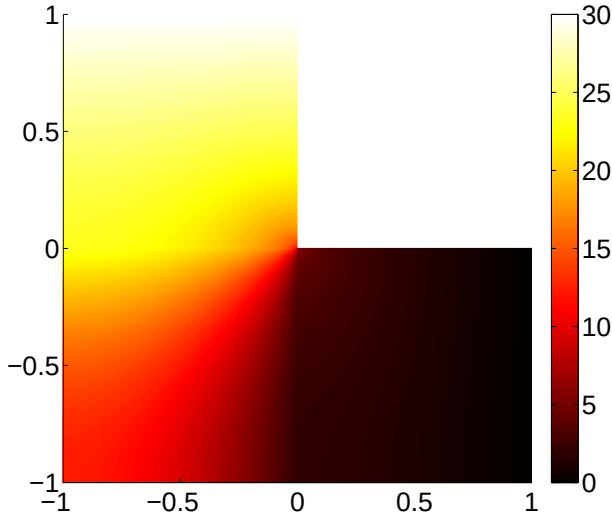


(a)

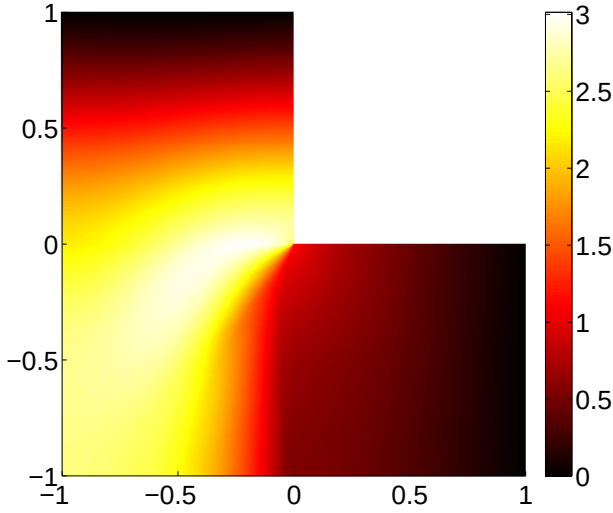


(b)

Figure 4.2: (a) The L-shaped domain. (b) Cross-section of the fuzzy solution along the dotted line in (a), with the gray-scale colors representing the membership value $\mu_{\widetilde{u(\boldsymbol{x})}}$.



(a)



(b)

Figure 4.3: (a) Core $[\widetilde{u}]_1$ of fuzzy solution. (b) Width of the fuzzy support $[\widetilde{u}]_0$.

of the error, which is given by

$$\|u^{\mathrm{r}}_{n_\phi} - u^{\mathrm{r}}_{n_\phi,n_\psi}\|_{C(\Xi;H^1_0(\Omega))}. \tag{4.22}$$

Here $u^{\mathrm{r}}_{n_\phi,n_\psi}$ is the computed response surface (3.102), and $u^{\mathrm{r}}_{n_\phi}$ is the finite element solution (3.95) of the parametric PDE (3.30). The $C(\Xi;H^1_0(\Omega))$-norm is approximated on the basis of a uniform quasi–Monte Carlo sampling of the parameter domain Ξ with 10^4 samples. The figures show the relative error

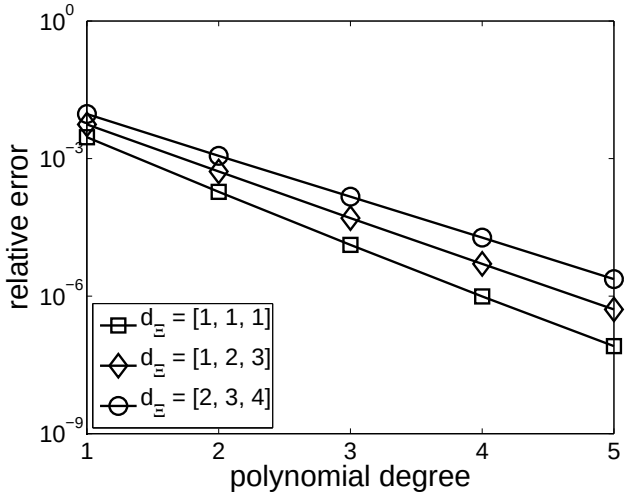
$$\frac{\|u^{\mathrm{r}}_{n_\phi} - u^{\mathrm{r}}_{n_\phi,n_\psi}\|_{C(\Xi;H^1_0(\Omega))}}{\|u^{\mathrm{r}}_{n_\phi}\|_{C(\Xi;H^1_0(\Omega))}}. \tag{4.23}$$

Fig. 4.4a and Fig. 4.5 show the error as a function of the polynomial degree for different numbers of fuzzy parameters d_Ξ, different correlation lengths L_c, and different σ/a_0. For all sets of the parameters, an exponential decay of the fuzzy error as a function of the polynomial degree is observed. The deviations of the exponential convergence in Fig. 4.5 for polynomial degree $p = 5$ are due to the relative accuracy of 10^{-9} in the Euclidian norm of the iterative solver. The difference in slope of the convergence graphs for different sets of parameters is caused by a difference in the width of the support of the fuzzy diffusion coefficient $\widetilde{a}$. Finally, Fig. 4.4b shows the error as a function of the number of polynomial basis functions ($n_\psi = \frac{(d_\Xi+p)!}{d_\Xi!p!}$). The convergence as a function of n_ψ turns out to be superalgebraic. The number of polynomial basis functions determines the size of the $\boldsymbol{G}_k$-matrices and is a good indicator for the amount of computational work (see §4.3.1 below).

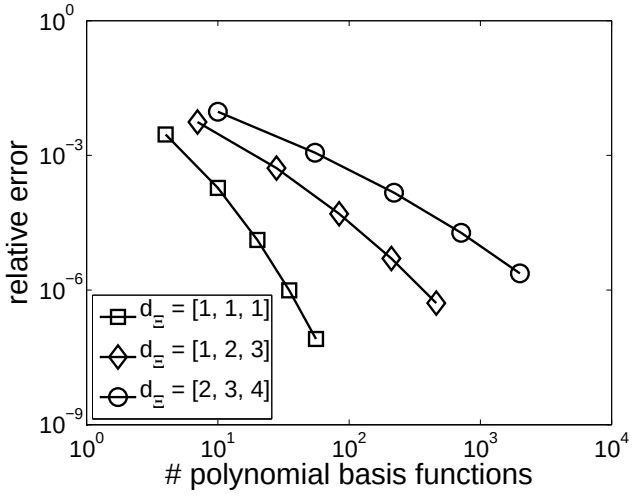
Convergence properties of iterative solvers

In this section, we compare the convergence properties and execution times of the iterative solvers proposed in §4.1 when applied to the fuzzy diffusion equation on the L-shaped domain. In the case of the center-based preconditioner (4.3), we approximate the inversion of $\boldsymbol{A}_0$ with one multigrid V(1,1)-cycle (i.e., $\gamma = 1$, $\nu_1 = \nu_2 = 1$ in Algorithm 1). Given an unstructured finite element discretization, we use Ruge–Stüben AMG [192] with Gauss–Seidel smoother. In the case of the collective smoothing multigrid method, we also consider the AMG variant and use this as preconditioner for the CG method. We apply W(2,1)-cycles (i.e., $\gamma = 2$, $\nu_1 = 2$, $\nu_2 = 1$ in Algorithm 1) with a block Jacobi smoother with damping factor 4/5. In both cases, the same AMG building blocks are used to construct the prolongation and coarse grid operators.

Table 4.4 shows the required number of iterations and solution time of CG preconditioned by either the center-based preconditioner or the collective

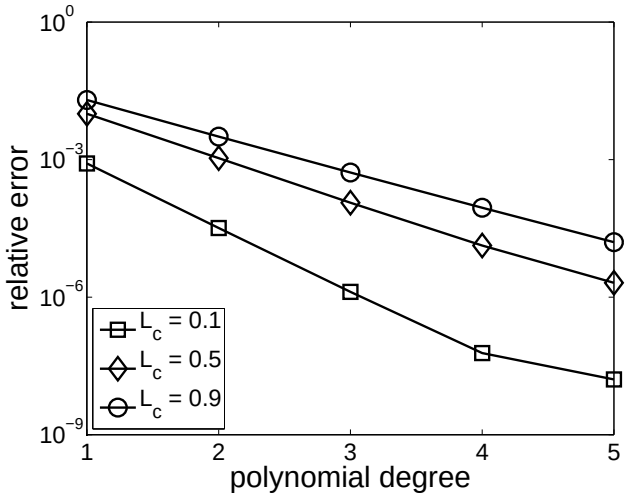


(a)

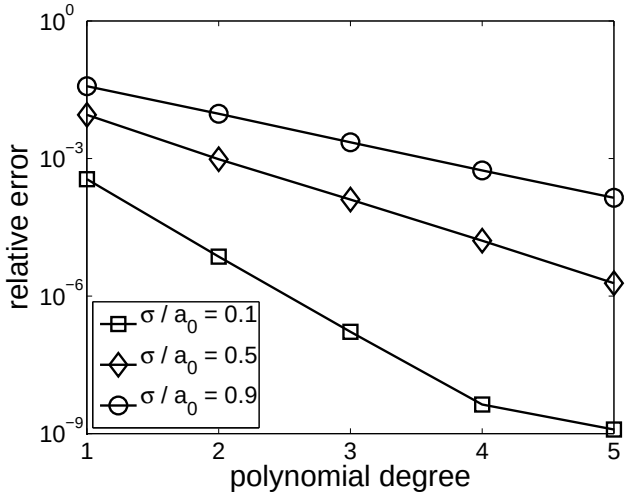


(b)

Figure 4.4: The relative error in the $C(\Xi; H_0^1(\Omega))$-norm of the polynomial response surface approximation of the solution to the parametric PDE (3.30) as (a) a function of the polynomial degree p and as (b) a function of the number of polynomial basis functions n_ψ for different numbers d_Ξ of fuzzy parameters ($n_\phi = 11\,258$, $L_c = [0.7, 0.3, 0.1]$, $\sigma/a_0 = [1/3,\ 0.4,\ 0.2]$).



(a)



(b)

Figure 4.5: The relative error in the $C(\Xi; H_0^1(\Omega))$-norm of the polynomial response surface approximation of the solution to the parametric PDE (3.30) as a function of the polynomial degree p for (a) different correlation lengths L_c and (b) different σ/a_0 ($n_\phi = 11\,258$, $d_\Xi = [1, 2, 3]$, and unless specified otherwise, $L_c = [0.7, 0.3, 0.1]$, $\sigma/a_0 = [1/3,\ 0.4,\ 0.2]$).

Table 4.4: Number of iterations and (in brackets) solution time in seconds for solving a fuzzy diffusion equation on the L-shaped domain depicted in Fig. 4.2a, with CG preconditioned by the center-based preconditioner or the collective smoothing multigrid approach. Unless specified otherwise, the following parameters are used: $n_\phi = 32\,499$, $d_\Xi = [1, 2, 3]$, $p = 3$, $n_\psi = 84$, $\sigma/a_0 = [1/3, 0.4, 0.2]$.

spatial nodes n_ϕ total DOF	$n_\phi = 32\,499$ (2 729 916)	$n_\phi = 64\,869$ (5 448 996)	$n_\phi = 127\,885$ (10 742 340)	$n_\phi = 257\,157$ (21 601 188)
CG-center-based	32 (35.7)	41 (99.0)	42 (250.1)	37 (495.2)
CG-AMG	14 (73.1)	15 (186.0)	14 (381.1)	15 (777.7)
fuzzy numbers d_Ξ total DOF	$d_\Xi = [1, 1, 1]$ (649 980)	$d_\Xi = [1, 2, 3]$ (2 729 916)	$d_\Xi = [2, 3, 4]$ (7 149 780)	$d_\Xi = [3, 4, 5]$ (14 787 045)
CG-center-based	32 (8.8)	32 (34.9)	33 (93.3)	35 (223.3)
CG-AMG	14 (16.9)	14 (70.7)	14 (234.6)	14 (578.1)
polynomial degree p total DOF	$p = 2$ (909 972)	$p = 3$ (2 729 916)	$p = 4$ (6 824 790)	$p = 5$ (15 014 538)
CG-center-based	31 (11.9)	32 (34.7)	32 (82.6)	32 (218.0)
CG-AMG	14 (28.3)	14 (73.3)	14 (174.5)	14 (455.8)
scaling factor σ/a_0	$\sigma/a_0 = 0.1$	$\sigma/a_0 = 0.3$	$\sigma/a_0 = 0.6$	$\sigma/a_0 = 0.9$
CG-center-based	29 (30.6)	31 (33.6)	36 (37.6)	40 (41.5)
CG-AMG	14 (71.1)	14 (70.1)	14 (70.7)	14 (72.9)

smoothing multigrid method for a variety of parameter settings. We observe for both preconditioners a robust convergence behavior with respect to the number of spatial nodes and the polynomial degree. The slight increase in iteration count for the center-based preconditioner in the case of a growing number of fuzzy numbers can be explained from the enlargement of the support of $\widetilde{a}$ in (3.28) when more—nonnegligible—expansion terms are taken into account. The effect of the width of the support is more clearly visible in the last row of Table 4.4, which shows an increase of iteration counts for the center-based preconditioner for large values of σ/a_0.

This table also presents solution times of the center-based and the collective smoothing multigrid preconditioner. Obviously, it is a delicate matter to compare timing results of two different codes, since they reflect not just algorithmic issues but also a multitude of implementation aspects. For what it is worth, we observe that both codes perform similarly overall. However, our current implementation of the center-based preconditioner appears to be faster than the collective smoothing multigrid implementation. This is mainly due to a lower set-up cost and a cheaper matrix-vector product.

4.3.2 Plane stress elasticity problem

Problem setup

As a second test case, we consider a plane stress elasticity problem. The geometry of the problem is illustrated in Fig. 4.6. It represents a two-dimensional plate with two clamped boundaries on the left side and one traction boundary on the right. All other boundaries are free. Denote by (u_{x_1}, u_{x_2}) the displacement vector describing the deformation of the material depicted in Fig. 4.6 under a load vector (f_{x_1}, f_{x_2}) and a boundary traction (t_{x_1}, t_{x_2}). Assuming isotropic and isothermal conditions, the displacement (u_{x_1}, u_{x_2}) is governed by

$$-\begin{bmatrix} \frac{\partial \cdot}{\partial x_1} & \frac{\partial \cdot}{\partial x_2} \end{bmatrix} \left(\sum_{i=1}^{2} \sum_{j=1}^{2} \boldsymbol{C}_{|\cdot,\cdot,i,j} \left(\begin{bmatrix} \frac{\partial \cdot}{\partial x_1} \\ \frac{\partial \cdot}{\partial x_2} \end{bmatrix} \begin{bmatrix} u_{x_1} & u_{x_2} \end{bmatrix} \right)_{|i,j} \right) = \begin{bmatrix} f_{x_1} & f_{x_2} \end{bmatrix}, \tag{4.24}$$

where $\boldsymbol{C}$ is the fourth-order stiffness tensor which is defined as

$$C_{1,1,\cdot,\cdot} = \begin{bmatrix} \frac{E}{1-\nu^2} & 0 \\ 0 & \frac{E\nu}{1-\nu^2} \end{bmatrix}, \qquad C_{1,2,\cdot,\cdot} = \begin{bmatrix} 0 & \frac{E}{2(1+\nu)} \\ \frac{E}{2(1+\nu)} & 0 \end{bmatrix},$$

$$C_{2,1,\cdot,\cdot} = C^T_{1,2,\cdot,\cdot}, \qquad C_{2,2,\cdot,\cdot} = \begin{bmatrix} \frac{E\nu}{1-\nu^2} & 0 \\ 0 & \frac{E}{1-\nu^2} \end{bmatrix}, \tag{4.25}$$

given the Young's modulus E and Poisson's ratio ν of the material. Equations (4.24)–(4.25) follow from the constitutive equations for the stress, i.e.,

$$\boldsymbol{\sigma} = \sum_{i=1}^{2}\sum_{j=1}^{2} \boldsymbol{C}_{|\cdot,\cdot,i,j}\boldsymbol{\epsilon}_{|i,j} \tag{4.26}$$

combined with the strain-displacement equations for the strain tensor $\boldsymbol{\epsilon}$

$$\boldsymbol{\epsilon} = \begin{bmatrix} \frac{\partial u_{x_1}}{\partial x_1} & \frac{1}{2}\left(\frac{\partial u_{x_1}}{\partial x_2} + \frac{\partial u_{x_2}}{\partial x_1}\right) \\ \frac{1}{2}\left(\frac{\partial u_{x_1}}{\partial x_2} + \frac{\partial u_{x_2}}{\partial x_1}\right) & \frac{\partial u_{x_2}}{\partial x_2} \end{bmatrix} \tag{4.27}$$

Assuming uncertain material parameters, we model the Young's modulus as a fuzzy field $\widetilde{E}$. As a consequence, we have a fuzzy displacement $(\widetilde{u}_{x_1}, \widetilde{u}_{x_2})$, a fuzzy strain $\widetilde{\boldsymbol{\epsilon}}$, and fuzzy stress $\widetilde{\boldsymbol{\sigma}}$.

In the numerical experiments, we consider a 1-by-1 meter aluminium plate with a thickness of 1 mm and a Poisson's ratio of $\nu = 0.35$. The right boundary is subjected to an outward oriented normal load of $5\,\mathrm{MN/m}$, corresponding to a surface traction of $(t_{x_1}, t_{x_2}) = (5000, 0)\,\mathrm{MN/m^2}$. The volume force is equal to $(f_{x_1}, f_{x_2}) = (0, -400)\,\mathrm{MN}$. The Young's modulus is given by a linear combination of d_Ξ noninteractive symmetrical triangular fuzzy numbers with support $[-1, 1]$ as in (3.28), where the deterministic functions E_k are constructed as $E_k = \sqrt{\kappa_k} v_k$ with κ_k and v_k, respectively, the eigenvalues and eigenfunctions of an exponential kernel $C(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp(-\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_1 / L_c)$ with $L_c = 1$ on a square. The deterministic function E_0 is taken equal to the Young's modulus of aluminium, $E_0 = 70 \cdot 10^3\,\mathrm{MN/m^2}$. Figures 4.7–4.9 illustrate the core, the width of the support, and a cross section of, respectively, the x_1 and x_2-components of the fuzzy displacement vector using $\sigma = 5 \cdot 10^{-2} E_0$, a spatial triangular mesh with $n_\phi = 59\,666$ degrees of freedom, $d_\Xi = 6$, and a Chebyshev polynomial response surface of degree $p = 3$. The cross section is taken at $x_1 = 0.25$.

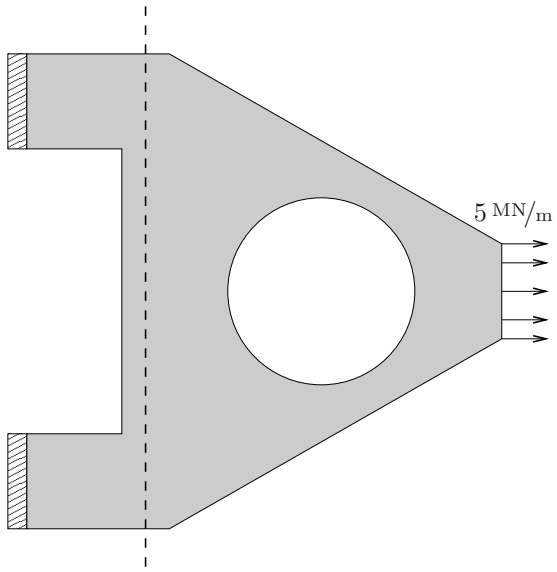


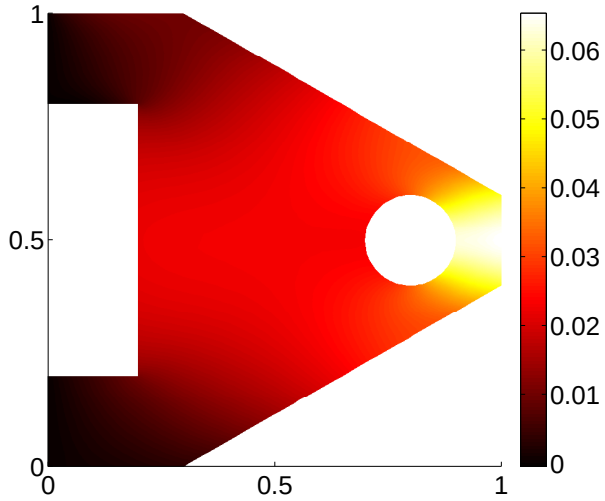
Figure 4.6: Geometry of plane stress elasticity problem. The dashed line indicates the position of the cross sections shown in Fig. 4.9.

Accuracy of spectral Galerkin approximation

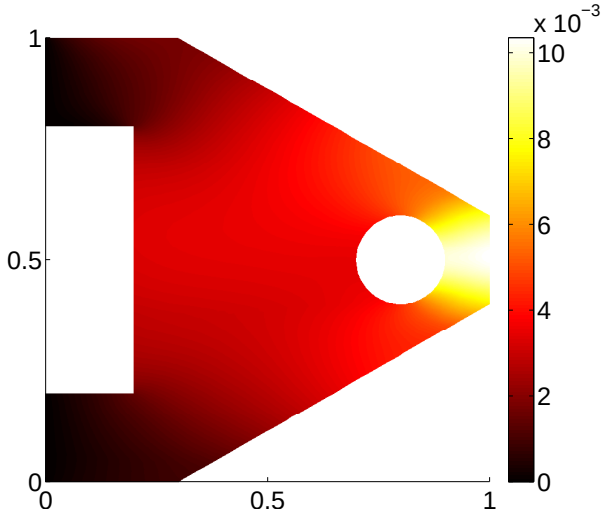
Similarly as for the fuzzy diffusion problem, we investigate the accuracy of the Chebyshev response surface approximation. To that end, we apply Corollary 3.2.1 again and measure the fuzzy error in the $C(\Xi; H_0^1(\Omega))$-norm (see (4.23)) using a uniform quasi–Monte Carlo sampling of the parameter domain Ξ with 10^4 samples. Fig. 4.10 and Fig. 4.11 show that for all variations of the parameters an exponential convergence of the fuzzy error as a function of the polynomial degree is obtained for the plane stress problem.

Convergence properties of iterative solvers

Although the iterative solvers proposed in §4.1 were originally developed for solving the Galerkin discretization of a scalar fuzzy PDE, they can easily be extended to the solution of a system of fuzzy PDEs, as in (4.24). In the case of a PDE system, the definition of the center-based preconditioner remains given by (4.3), but the multigrid components of the collective smoothing multigrid method described in §4.1.2 require modifications. These modifications are also

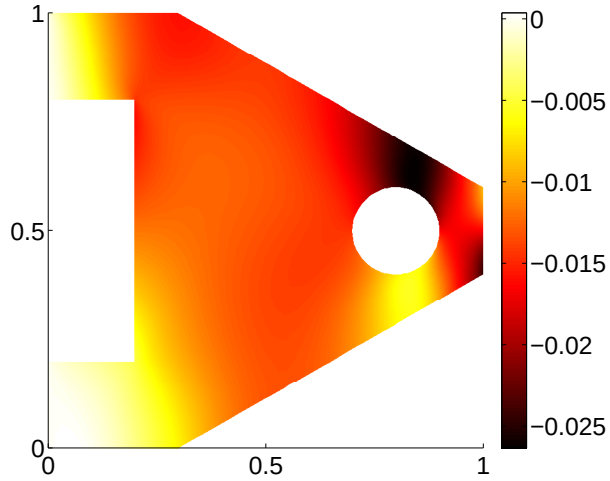


(a)

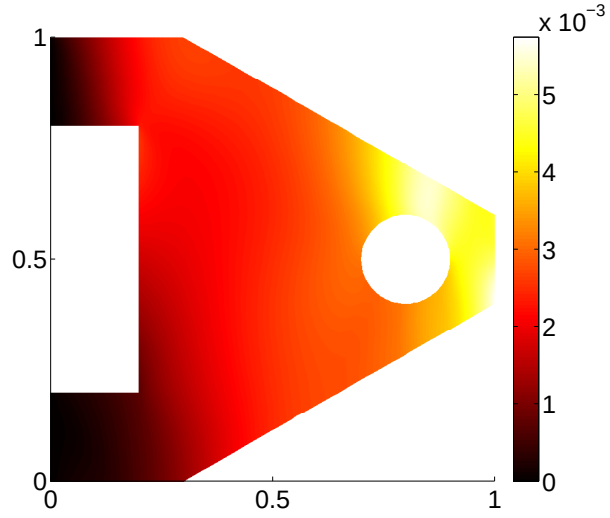


(b)

Figure 4.7: (a) Core and (b) width of the support of the first component $\widetilde{u}_{x_1}$ of the fuzzy displacement vector of the fuzzy plane stress problem.

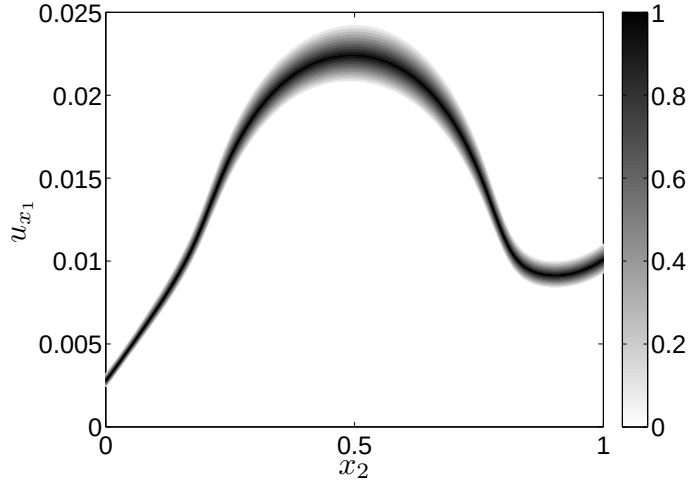


(a)

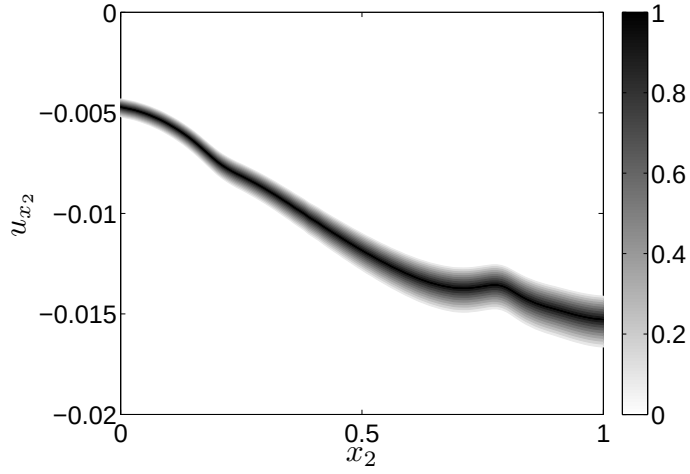


(b)

Figure 4.8: (a) Core and (b) width of the support of the second component $\widetilde{u}_{x_2}$ of the fuzzy displacement vector of the fuzzy plane stress problem.

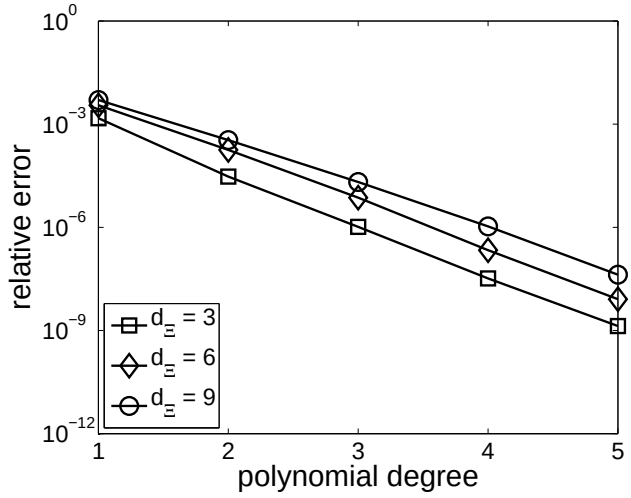


(a)

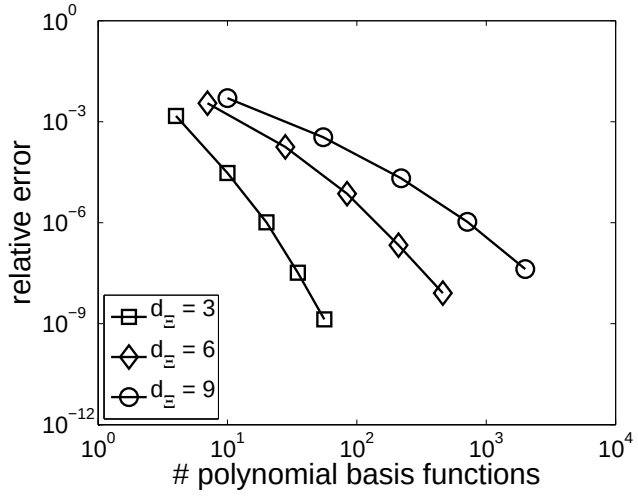


(b)

Figure 4.9: Vertical cross section of (a) the first component $\widetilde{u}_{x_1}$ and (b) the second component $\widetilde{u}_{x_2}$ of the fuzzy displacement vector of the fuzzy plane stress problem.

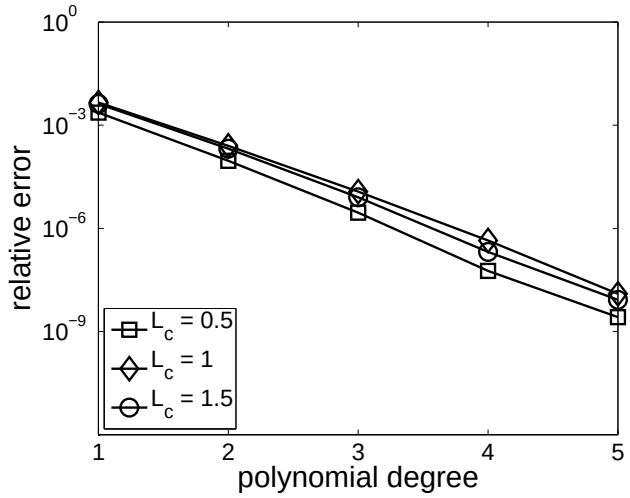


(a)

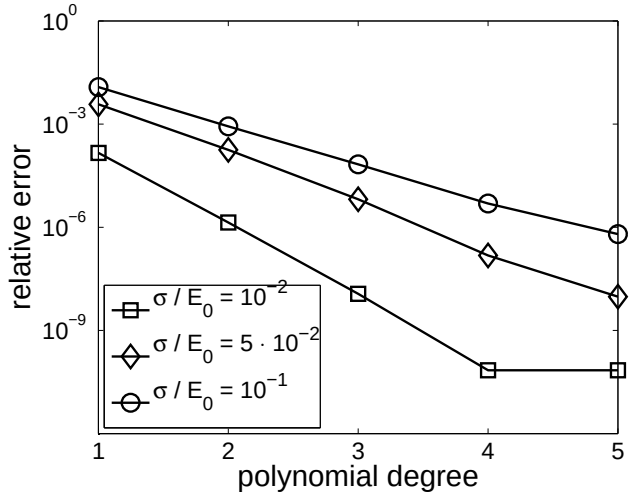


(b)

Figure 4.10: The relative error in the $C(\Xi; H_0^1(\Omega))$-norm of the polynomial response surface approximation of the solution to the fuzzy plane stress problem (4.24) as (a) a function of the polynomial degree p and as (b) a function of the number of polynomial basis functions n_ψ ($n_\phi = 12\,732$, $L_c = 1$, $\sigma/E_0 = 5 \cdot 10^{-2}$).



(a)



(b)

Figure 4.11: The relative error in the $C(\Xi; H_0^1(\Omega))$-norm of the polynomial response surface approximation of the solution to the fuzzy plane stress problem (4.24) as a function of the polynomial degree p for (a) different correlation lengths L_c and (b) different supports σ/E_0 ($n_\phi = 12\,732$, $d_\Xi = 6$, and unless specified otherwise, $L_c = 1$, $\sigma/E_0 = 5 \cdot 10^{-2}$).

needed for the multigrid solver that is used to invert the $\boldsymbol{A}_0$-matrix in the center-based preconditioner (4.3) approximately.

Applying AMG for scalar PDEs straightforwardly to PDE systems is known to often result in a performance degradation [197]. Typical multigrid solutions for PDE systems include the unknown-based and point-based multigrid approach [192]. These multigrid methods apply a blockwise construction of the prolongation and coarse grid operators, and possibly also of the relaxation operators. The block size is determined either by the number of degrees of freedom per physical unknown in the unknown-based case, or by the number of unknowns per spatial node in the point-based case. An additional multigrid difficulty arises in the context of elasticity problems. Classical AMG interpolation operators do not adequately represent the null-space of the corresponding PDE operator, which consists of the so-called rigid body modes. In the case of a two-dimensional plane stress problem, the rigid body modes are composed of two translations and one rotation vector. The translations are captured in the coarse grid operator when using classical (Ruge–Stüben) AMG, but the rotation is not. As a result, the convergence of unknown-based classical AMG deteriorates in the case that the system is nearly singular, i.e., when Dirichlet boundary conditions are imposed on only a small part of the domain. To resolve this issue, several solutions have been proposed. We follow the approach detailed in [200], where a point-based smoothed aggregation AMG is proposed that makes use of the prior knowledge of the rigid body modes.

The smoothed aggregation AMG setup is used both as part of the center-based preconditioner and for constructing the multigrid hierarchy for the collective smoothing multigrid preconditioner. That is, the inversion of the center-based matrix $\boldsymbol{A}_0$ in the center-based preconditioner is approximated by a V(1,1)-AMG cycle (i.e., $\gamma = 1$, $\nu_1 = \nu_2 = 1$ in Algorithm 1) with Gauss–Seidel smoother and smoothed aggregation AMG interpolation matrices. The collective smoothing multigrid method applies W(2,2)-cycles (i.e., $\gamma = 2$, $\nu_1 = \nu_2 = 2$ in Algorithm 1) with a collective Jacobi smoother with a damping factor of 2/3.

Table 4.5 illustrates the convergence of the methods. We note that similar optimal convergence properties are observed as for the fuzzy diffusion problem: a convergence rate that is independent of the size of the spatial and fuzzy discretization parameters. The center-based preconditioner is, however, in contrast to the multigrid method, not robust with respect to the width of the support of the fuzzy input. The computational cost of the collective smoothing method is again overall higher than the cost of the center-based preconditioner.

Table 4.5: Number of iterations and (in brackets) solution time in seconds for solving the fuzzy plane stress problem on the domain depicted in Fig. 4.6, with CG preconditioned by the center-based preconditioner or the collective smoothing multigrid approach. Unless specified otherwise, the following parameters are used: $n_\phi = 59\,666$, $d_\Xi = 3$, $p = 2$, $n_\psi = 10$, $\sigma/E_0 = 10^{-2}$.

spatial nodes n_ϕ total DOF	$n_\phi = 59\,666$ (596 660)	$n_\phi = 108\,828$ (1 088 280)	$n_\phi = 160\,124$ (1 601 240)	$n_\phi = 203\,370$ (2 033 700)
CG-center-based	46 (16.9)	48 (33.0)	48 (49.6)	50 (73.0)
CG-AMG	28 (33.5)	29 (64.4)	29 (100.4)	29 (134.8)
fuzzy numbers d_Ξ total DOF	$d_\Xi = 1$ (178 998)	$d_\Xi = 3$ (596 660)	$d_\Xi = 6$ (1 670 648)	$d_\Xi = 9$ (3 281 630)
CG-center-based	45 (7.8)	46 (18.3)	46 (43.7)	46 (93.9)
CG-AMG	28 (11.6)	28 (33.9)	28 (111.1)	28 (268.0)
polynomial degree p total DOF	$p = 1$ (238 664)	$p = 3$ (1 193 320)	$p = 4$ (2 088 310)	$p = 5$ (3 341 296)
CG-center-based	45 (10.1)	46 (27.1)	46 (44.1)	46 (66.5)
CG-AMG	28 (19.9)	28 (56.3)	28 (93.8)	28 (148.8)
scaling factor σ/E_0	$\sigma/E_0 = 10^{-3}$	$\sigma/E_0 = 0.01$	$\sigma/E_0 = 0.1$	$\sigma/E_0 = 0.5$
CG-center-based	45 (18.4)	46 (17.9)	50 (20.2)	68 (27.4)
CG-AMG	28 (39.0)	28 (38.1)	28 (37.9)	28 (40.0)

4.4 Conclusions

This chapter presented a numerical study and analysis of a spectral Galerkin method for solving fuzzy PDEs. In the literature, the response surfaces for solving fuzzy PDEs are typically constructed by means of a sparse grid interpolation or very generic response surface techniques like Kriging. Here, we demonstrated that spectral Galerkin methods, originally created for stochastic PDEs, can also be applied very effectively for solving fuzzy PDEs.

In the fuzzy context, the accuracy of an approximation is measured in quite a different norm than in the stochastic context. This leads us to use Chebyshev expansions rather than the more classical Hermite or Legendre expansions that are used for stochastic PDEs. This Chebyshev representation turns out to yield very accurate (in the fuzzy sense) approximations. We numerically demonstrated exponential convergence as a function of the polynomial degree for two nontrivial example problems. The convergence properties of the two preconditioners developed in this chapter were shown to be optimal w.r.t. the discretization parameters. This was shown both theoretically, by means of a local Fourier mode analysis, as well as numerically, by means of an extensive set of numerical experiments. A numerical convergence study on a fuzzy diffusion problem and a fuzzy elasticity problem showed that both methods perform quite well, also on problems defined on irregular meshes and for systems of equations. The center-based preconditioner turned out to be the most efficient solver overall. On the other hand, the collective smoothing multigrid outperforms the center-based preconditioner w.r.t. robustness, by showing an almost constant number of iterations over a very wide range of parameter values.

Chapter 5

Low-rank tensor based methods for fuzzification

In this chapter, we turn to the problem of computing $f(\widetilde{\boldsymbol{\xi}})$ by the α-cut approach, where $\widetilde{\boldsymbol{\xi}} := (\widetilde{\xi}_1, \dots \widetilde{\xi}_d) \in \mathcal{F}(\mathbb{R}^d)$ is a normal and compact fuzzy set and $f\colon \mathbb{R}^d \supset [\widetilde{\boldsymbol{\xi}}]_0 \to \mathbb{R}$ is a continuous function. Furthermore, we assume that $\widetilde{\xi}_1, \dots \widetilde{\xi}_d$ are noninteractive. According to Theorem 2.2.1, the α-cut approach then leads to a sequence of global minimization and global maximization problems over nested hyperrectangles. We propose a derivative-free optimization algorithm which combines the low-rank tensor approximation algorithm from [9] with an algorithm from [71, 127]. The results of this chapter have been submitted for publication to *Fuzzy Sets and Systems*.

5.1 Introduction

The curse of dimensionality [16, 188] poses a big challenge in the global optimization of continuous functions over a hyperrectangular search space. Without extra assumptions on the structure of the function, the problem is known to scale exponentially in size with the dimension. A common extra assumption is that f is Lipschitz continuous. In this special case, it is known that no derivative-free optimization algorithm can do better than grid search, i.e., choosing the optimal function value from function values obtained by sampling the function on an equidistant tensor grid [164, 201]. Hence, the assumption that f is Lipschitz continuous is not enough to break the curse

of dimensionality. Grid search will, nevertheless, be the starting point of our algorithm.

In many practical problems, the tensor generated by a tensor grid sampling can be well approximated by a low-rank tensor [113]. Therefore, we propose a fuzzification algorithm which starts with the construction of a low-rank (hierarchical Tucker) tensor approximation of the function on this tensor grid using the algorithm of [9]. After the construction of the low-rank approximation, the proposed fuzzification algorithm performs the actual optimization. This is done using a method from [71, 127]. This method returns an approximation of the minimal and the maximal entry in the low-rank tensor. In Section 5.3.3, we derive an error and complexity estimate for the proposed fuzzification algorithm. Finally, we end the chapter with a comparison of our fuzzification algorithm and some state-of-the-art global optimization routines in a numerical test with some challenging high-dimensional optimization problems.

5.2 Fuzzification of functions

5.2.1 The α-cut approach

For a normal and compact fuzzy set $\widetilde{\boldsymbol{\xi}} := (\widetilde{\xi}_1, \ldots, \widetilde{\xi}_d) \in \mathcal{F}(\mathbb{R}^d)$ and a continuous function $f\colon \mathbb{R}^d \supset [\widetilde{\boldsymbol{\xi}}]_0 \to \mathbb{R}$, we have, according to Theorem 2.2.1, that

$$[f(\widetilde{\boldsymbol{\xi}})]_\alpha = [\min_{\boldsymbol{\xi} \in [\widetilde{\boldsymbol{\xi}}]_\alpha} f(\boldsymbol{\xi}), \max_{\boldsymbol{\xi} \in [\widetilde{\boldsymbol{\xi}}]_\alpha} f(\boldsymbol{\xi})]. \tag{5.1}$$

A possible numerical approach to compute $f(\widetilde{\boldsymbol{\xi}})$ is to start with the computation of a fixed selection of α-cuts (typically equidistant) and further refine this selection to achieve convergence in some sense (the intermediate α-cuts are most often simply computed by a piecewise linear interpolation). This is the so-called α-cut approach.

For each α-level considered, the global minimum and the global maximum of f over $[\widetilde{\boldsymbol{\xi}}]_\alpha$ have to be computed. As such, the α-cut approach leads to a sequence of global optimization problems over nested search spaces. We will confine ourselves to fuzzy sets $\widetilde{\boldsymbol{\xi}}$ with noninteractive entries $\widetilde{\xi}_1, \ldots, \widetilde{\xi}_d$. In such case, the search spaces $[\widetilde{\boldsymbol{\xi}}]_\alpha$ are hyperrectangles.

The error which is introduced by the numerical approximation of the α-cuts can be quantified easily in the supremum distance (2.6). Let $f_{\mathrm{appr}}(\widetilde{\boldsymbol{\xi}})$ denote the numerical approximation of $f(\widetilde{\boldsymbol{\xi}})$ obtained through the α-cut approach. Then,

by Theorem 2.4.1, we have that

$$\epsilon_{\text{appr}} := d_\infty(f(\widetilde{\boldsymbol{\xi}}), f_{\text{appr}}(\widetilde{\boldsymbol{\xi}})) \tag{5.2}$$

$$= \sup_{0\leq\alpha\leq 1} \max(|[f(\widetilde{\boldsymbol{\xi}})]_\alpha^- - [f_{\text{appr}}(\widetilde{\boldsymbol{\xi}})]_\alpha^-|, |[f(\widetilde{\boldsymbol{\xi}})]_\alpha^+ - [f_{\text{appr}}(\widetilde{\boldsymbol{\xi}})]_\alpha^+|).$$

Under the simplifying assumption that every α-cut is computed by a global minimization and a global maximization, we can conclude that the error produced by the numerical solution of these optimization problems straightforwardly translates to the error ϵ_{appr} in the fuzzy sense.

5.2.2 Global optimization

The complexity of global optimization

The design and the performance of an algorithm for the global optimization (minimization or maximization) of a continuous function f over the hyperrectangle $[\widetilde{\boldsymbol{\xi}}]_\alpha$ depends on the a priori knowledge of the function and on the further information that can be retrieved from the function. A typical example of a priori knowledge is that the function is known to be a member of a certain function class. The function may, for example, be known to be Lipschitz continuous. Further information that can be retrieved from the function may, for example, be its function value $f(\boldsymbol{\xi})$ for arbitrary $\boldsymbol{\xi}$. This is the so-called derivative-free optimization setting [156, 179].

A typical way of assessing the performance of an algorithm is to measure its worst-case complexity behavior over all possible functions that satisfy the a priori knowledge to achieve a certain accuracy. This depends, of course, on the precise definition of complexity. In the framework of derivative-free optimization problems, complexity is often defined as the number of function evaluations. This is the so-called information-based complexity [32, 49, 98]. The use of information-based complexity may, however, be deceptive.

The information-based complexity of the optimization problem is bounded from above by the information-based complexity of the approximation problem in the $C([\widetilde{\boldsymbol{\xi}}]_\alpha)$-norm [164]. Suppose, for example, that we were able to construct an interpolating polynomial approximation f_{appr} of f using N function evaluations. Because

$$|\min_{\boldsymbol{\xi}\in[\widetilde{\boldsymbol{\xi}}]_\alpha} f(\boldsymbol{\xi}) - \min_{\boldsymbol{\xi}\in[\widetilde{\boldsymbol{\xi}}]_\alpha} f_{\text{appr}}(\boldsymbol{\xi})| \leq \left\|f - f_{\text{appr}}\right\|_{C([\widetilde{\boldsymbol{\xi}}]_\alpha)} \tag{5.3}$$

and because the global optimum of a polynomial can be computed exactly (using, e.g., the sums-of-squares approach [130, 131, 169, 170, 171]), we can thus

approximate the global optimum of f with a same error as the approximation problem using N function evaluations.

This is, of course, misleading. The problem of optimizing f is transferred to the problem of optimizing the polynomial f_{appr}. The information-based complexity does not cover the computational cost of optimizing f_{appr}, for which no efficient algorithm may be known. As to the sums-of-squares approach to polynomial optimization, for example, there are only few results known about its computational cost [50, 132, 142, 161, 184]. A *computational* complexity model is then more appropriate to express the real cost of solving the optimization problem. The black-box model (an extension of the real number model), for example, counts both the number of function evaluations and the number of arithmetic operations. In the case of Lipschitz continuous functions, we have, however, the important result that no derivative-free optimization algorithm can do better than grid search in terms of the worst-case information-based complexity as well as the computational complexity. For a more elaborate discussion on computational models and complexity in the optimization framework, we refer to [49].

Practical algorithms

The class of Lipschitz continuous functions is a rather broad class. In practical problems, the function to optimize can generally be assumed to have some specific structure. The knowledge of optimal algorithms and the performance of concrete algorithms for such functions with a specific structure can, however, be very hard to obtain. An easier and quicker approach to the design of an optimization algorithm and the analysis of its performance is to do a numerical test with different classes of test functions. It allows to quickly spot the strengths and weaknesses of a new algorithm in comparison with other standard algorithms. We will follow this approach and do an extensive numerical test of our algorithm in Section 5.4.

A minimal (and often easy to achieve) requirement that is, however, generally expected of an optimization algorithm is completeness [158]. An optimization algorithm is called complete if it is sure to converge to the global optimum regardless of a specific rate of convergence. When the function is known to be continuous, completeness is, for example, guaranteed if the algorithm generates an increasingly dense sampling of the search space.

We give here an overview of some of the ingredients that are commonly found in derivative-free optimization algorithms which are applicable to the optimization of a continuous function over a hyperrectangular search space.

- **branching:** The hyperrectangular search space is hierarchically subdivided in smaller and smaller boxes. Boxes which are found to be potential candidates for containing the global optimum are marked for splitting, other boxes are temporarily marked as uninteresting. Keeping a balance between global and local search is a key issue in branching methods. Examples of algorithms with a branching strategy can be found in [32, 102, 107].
- **global or local approximation:** A global [85, 99, 108, 106, 178] or local [32, 102] (e.g., in the boxes created in the branching strategy) approximation of the function is constructed. This approximation can be a radial basis approximation [85, 178], a Kriging approximation [99, 108], a polynomial [32] approximation, etcetera. The approximation is used to steer the sampling of the function towards more interesting areas. These samples can then on their turn be used to update the approximations. Information from the approximation can, for example, be derived by:
  - optimization: A global or local optimization of the response surface approximation is performed. Any method which is applicable may be used.
  - relaxation: This is a form of approximate optimization. An upper and a lower bound to the global optima of the approximation may, for example, be computed using interval computations [89, 90].
- **utility function:** A utility function is constructed together with the global or local approximation. Such utility function can, for example, be an expected improvement function [99, 106]. This is a function which gives an estimate of the local error of the approximation. This can improve the quality of the information retrieved from the approximation.
- **model assumption:** A model assumption can be used to decide on the type of global or local approximation. It can also be used directly. Based on existing samples of the function and an estimated Lipschitz constant, for example, some regions can be temporarily excluded from the global search [98, 107].
- **randomization:** Stochastic optimization of some sort of the function or the approximations [191]. Examples of these methods are: multistart methods, particle swarm optimization, simulated annealing, genetic programming, Monte Carlo sampling, Latin Hypercube sampling.
- **local optimization:** Local optimization can be used for the optimization of a local approximation or the function itself [102]. It is, for example, an essential component in multistart methods. A multistart method starts

with a random sampling of the search space and then initiates a local optimization procedure from each of these samples.

5.2.3 Global optimization in the α-cut approach

The α-cut approach leads to a sequence of global optimization problems over nested search spaces. As such, it can be assumed that a significant reduction of the computational cost can be achieved if some information is recycled from one optimization problem to the next optimization problem. Information that can be recycled is for example: function values, global or local approximations, the global optima or bounds on these optima over certain regions in the search space (in a branching scheme for example), etcetera.

The order in which the optimization problems are solved can be of considerable importance. The most common ordering in the case that a fixed number of α-cuts is computed is the one where the optimization problems are solved from highest α-level to lowest α-level. Another important factor in the recycling of information is the curse of dimensionality. If the search spaces scale proportionally in each coordinate direction, then the ratio of the size of the search space on two different α-levels scales exponentially with the dimension. The information that two α-levels share thus quickly diminishes with increasing dimension.

Some examples of global optimization methods applied to the fuzzification of functions that can be found in the literature are: stochastic methods [153, 183], a branching scheme using local spline approximations [24], a global response surface approximation based on a sparse grid polynomial interpolation followed by different optimization methods [120, 121], the Gradual α-level Decreasing method [52] (this is a trajectory method), Latin Hypercube sampling [67, 97]. A widely applied fuzzification method is the vertex method [59, 136, 217], also called the Fuzzy Weighted Algorithm. It was generalized in [91] to the so-called Transformation Method. It is a non-adaptive derivative-free fuzzification algorithm which chooses the optimal function value from function values that are obtained by a sampling on the boundaries of the search spaces. Some adaptive variants of the Transformation Method can be found in [58, 92].

5.3 A low-rank tensor approximation based fuzzification technique

Grid search is the starting point of our two-stage algorithm. In the first stage, a low-rank tensor approximation of this tensor grid sampling of the function is constructed. Such a low-rank approximation could come from various sources; see, e.g., Section 3.3.4. To stay in the framework of derivative-free optimization and to keep things as general as possible, we use the black box approximation method as proposed in [9]. It is described in Section 5.3.3. The second stage consists of an algorithm which searches the maximal and the minimal entries in the low-rank tensor approximation [71, 127]. This is the actual optimization step. It is described in Section 5.3.4. Exchange of information between the different optimization problems in the α-cut approach is solely established by the shared construction of the low-rank tensor approximation.

5.3.1 Grid sampling approach

We start with a description of the grid sampling method, and focus in particular on the error incurred by that approach. We assume that the function f is Lipschitz continuous on the support of $\widetilde{\boldsymbol{\xi}}$, i.e., there exists a positive constant C such that

$$|f(\boldsymbol{\xi}_2) - f(\boldsymbol{\xi}_1)| \leq C|\boldsymbol{\xi}_2 - \boldsymbol{\xi}_1|, \quad \boldsymbol{\xi}_1, \boldsymbol{\xi}_2 \in [\widetilde{\boldsymbol{\xi}}]_0\,. \tag{5.4}$$

Let S_k be a set of n_k equidistantly distributed sample points in the kth dimension, i.e.,

$$S_k := \left\{ s_{k,i_k} := [\widetilde{\xi}_k]_0^- + (i_k - 1)h_k \right\}_{i_k=1}^{n_k} \quad \text{with} \quad h_k := \frac{[\widetilde{\xi}_k]_0^+ - [\widetilde{\xi}_k]_0^-}{n_k - 1}\,.$$

Evaluating f on the tensor grid $S_1 \times \cdots \times S_d$, yields a tensor $\boldsymbol{B} \in \mathbb{R}^{\mathcal{I}_{\boldsymbol{n}}}$ with

$$B_{\boldsymbol{i}} := f(s_{1,i_1}, \ldots, s_{d,i_d}), \quad \boldsymbol{i} \in \mathcal{I}_{\boldsymbol{n}}, \tag{5.5}$$

where $\mathcal{I}_{\boldsymbol{n}}$ is defined as

$$\mathcal{I}_{\boldsymbol{n}} := \bigtimes_{k=1}^{d} \mathcal{I}_{n_k} \quad \text{with} \;\; \mathcal{I}_n := \{1, \ldots, n\}. \tag{5.6}$$

Computing the α-cuts of $f(\widetilde{\boldsymbol{\xi}})$ now boils down to a search for the minimal and maximal element in a part of $\boldsymbol{B}$:

$$[f(\widetilde{\boldsymbol{\xi}})]_\alpha \approx [f_{\text{grid}}(\widetilde{\boldsymbol{\xi}})]_\alpha := [\min_{\substack{\boldsymbol{i} \in \mathcal{I}_{\boldsymbol{n}} \\ (s_{1,i_1}, \ldots, s_{d,i_d}) \in [\widetilde{\boldsymbol{\xi}}]_\alpha}} B_{\boldsymbol{i}}, \max_{\substack{\boldsymbol{i} \in \mathcal{I}_{\boldsymbol{n}} \\ (s_{1,i_1}, \ldots, s_{d,i_d}) \in [\widetilde{\boldsymbol{\xi}}]_\alpha}} B_{\boldsymbol{i}}]. \tag{5.7}$$

From (5.4), we can derive that the exact bounds of $[f(\widetilde{\boldsymbol{\xi}})]_\alpha$ for all α-cuts will differ at most $\frac{C}{2}\sqrt{\sum_{k=1}^{d} h_k^2}$ from the bounds of $[f_{\text{grid}}(\widetilde{\boldsymbol{\xi}})]_\alpha$. Hence, the error produced by a full tensor grid sampling optimization approach is bounded as

$$\epsilon_{\text{grid}} := d_\infty(f(\widetilde{\boldsymbol{\xi}}), f_{\text{grid}}(\widetilde{\boldsymbol{\xi}})) \leq \frac{C}{2}\sqrt{\sum_{k=1}^{d} h_k^2} = \mathcal{O}(\frac{\sqrt{d}}{\min_{k=1,\ldots,d} n_k}); \tag{5.8}$$

see also [164, 201].

In the simple case that $n_1 = \cdots = n_d$, this leads to a convergence rate of $\mathcal{O}(N_{\text{eval}}^{-\frac{1}{d}})$, where $N_{\text{eval}} = \prod_{k=1}^{d} n_k$. For higher dimensions $d \gg 1$, this is of course prohibitively expensive. Fortunately, it turns out that the tensor $\boldsymbol{B}$ in (5.5) generated by a function can, in certain cases, be approximated quite well by a low-rank tensor $\boldsymbol{A} \approx \boldsymbol{B}$ [113]. The complexity of constructing and storing such a low-rank tensor can be significantly smaller than that of constructing and storing $\boldsymbol{B}$. We will discuss such an algorithm in the next section. However, first we will quantify the error on the fuzzification result incurred by replacing $\boldsymbol{B}$ in (5.7) by its approximation $\boldsymbol{A}$. Setting

$$[f_{\text{trunc}}(\widetilde{\boldsymbol{\xi}})]_\alpha := [\min_{\substack{\boldsymbol{i}\in\mathcal{I}_{\boldsymbol{n}} \\ (s_{1,i_1},\ldots,s_{d,i_d})\in[\widetilde{\boldsymbol{\xi}}]_\alpha}} A_{\boldsymbol{i}}, \max_{\substack{\boldsymbol{i}\in\mathcal{I}_{\boldsymbol{n}} \\ (s_{1,i_1},\ldots,s_{d,i_d})\in[\widetilde{\boldsymbol{\xi}}]_\alpha}} A_{\boldsymbol{i}}], \tag{5.9}$$

the error of this low-rank approach can be bounded as

$$d_\infty(f(\widetilde{\boldsymbol{\xi}}), f_{\text{trunc}}(\widetilde{\boldsymbol{\xi}})) \leq d_\infty(f(\widetilde{\boldsymbol{\xi}}), f_{\text{grid}}(\widetilde{\boldsymbol{\xi}})) + d_\infty(f_{\text{grid}}(\widetilde{\boldsymbol{\xi}}), f_{\text{trunc}}(\widetilde{\boldsymbol{\xi}})), \tag{5.10}$$

where the extra error term $\epsilon_{\text{trunc}} := d_\infty(f_{\text{grid}}(\widetilde{\boldsymbol{\xi}}), f_{\text{trunc}}(\widetilde{\boldsymbol{\xi}}))$ can be bounded as

$$\begin{aligned}\epsilon_{\text{trunc}} = \sup_{0\leq\alpha\leq 1} \max\Big(&|[f_{\text{grid}}(\widetilde{\boldsymbol{\xi}})]_\alpha^+ - [f_{\text{trunc}}(\widetilde{\boldsymbol{\xi}})]_\alpha^+|, \\ &|[f_{\text{grid}}(\widetilde{\boldsymbol{\xi}})]_\alpha^- - [f_{\text{trunc}}(\widetilde{\boldsymbol{\xi}})]_\alpha^-|\Big) \\ \leq \max_{\boldsymbol{i}\in\mathcal{I}_{\boldsymbol{n}}} &|B_{\boldsymbol{i}} - A_{\boldsymbol{i}}| \equiv \|\boldsymbol{B} - \boldsymbol{A}\|_\infty.\end{aligned} \tag{5.11}$$

Hence, we will be interested in constructing a low-rank approximation which is accurate in the maximum norm.

The search for the minimal and maximal elements in parts of $\boldsymbol{A}$ by the approach (5.9) seems to require that every individual element $A_{\boldsymbol{i}}$ is constructed. If so, that would obviously ruin an overall low time complexity of the low-rank

tensor approach. An algorithm is therefore needed which can compute the extremal elements of a low-rank tensor in a cheap way, preferably in the same time complexity as the low-rank tensor construction. In the next section, we will discuss a possible candidate. Letting $\min_{\text{appr}}$ and $\max_{\text{appr}}$ be the operators that represent the algorithms that return an approximation of the minimal and maximal entry in a low-rank tensor, we have that the final approximation $[f_{\text{appr}}(\widetilde{\boldsymbol{\xi}})]_\alpha$ of $[f(\widetilde{\boldsymbol{\xi}})]_\alpha$ is equal to

$$[f_{\text{opt}}(\widetilde{\boldsymbol{\xi}})]_\alpha := [\min_{\substack{\boldsymbol{i}\in\mathcal{I}_{\boldsymbol{n}} \\ (s_{1,i_1},\dots,s_{d,i_d})\in[\widetilde{\boldsymbol{\xi}}]_\alpha}}{}^{\text{appr}} A_{\boldsymbol{i}}, \max_{\substack{\boldsymbol{i}\in\mathcal{I}_{\boldsymbol{n}} \\ (s_{1,i_1},\dots,s_{d,i_d})\in[\widetilde{\boldsymbol{\xi}}]_\alpha}}{}^{\text{appr}} A_{\boldsymbol{i}}]. \tag{5.12}$$

To sum up, the total error of the black box low-rank approximation approach can be bounded by the sum of three error terms. Defining $\epsilon_{\text{opt}} := d_\infty(f_{\text{trunc}}(\widetilde{\boldsymbol{\xi}}), f_{\text{opt}}(\widetilde{\boldsymbol{\xi}}))$, we have

$$\epsilon_{\text{appr}} \leq \epsilon_{\text{grid}} + \epsilon_{\text{trunc}} + \epsilon_{\text{opt}}\,. \tag{5.13}$$

5.3.2 A low-rank tensor format

Recently, there has been considerable interest in the research of various low-rank tensor formats. We refer to [84, 113, 124] for an overview. Very common formats are the canonical polyadic (CP) decomposition [23, 93] and the Tucker decomposition [198]. These two formats are extremes on a spectrum of tensor formats which can be represented by a tensor network [100]. The $\mathcal{H}$-Tucker format [83, 88] which we will use, can in fact be represented by a tensor network with a tree structure and is based on the notion of tensor *matricizations*.

Let t' denote the complement of a collection of indices $t \subset \{1,\dots,d\}$, i.e.,

$$t' := \{1,\dots,d\} \setminus t. \tag{5.14}$$

The matricizations of a tensor are then defined as follows.

Definition 5.3.1 (matricization)**.** *For a tensor $\boldsymbol{A} \in \mathbb{R}^{\mathcal{I}_{\boldsymbol{n}}}$ and a collection of dimension indices $t \subset \{1,\dots,d\}$, the matricization $\boldsymbol{A}^{(t)} \in \mathbb{R}^{\mathcal{I}_{\boldsymbol{n}|t} \times \mathcal{I}_{\boldsymbol{n}|t'}}$ is defined by its entries as*

$$A^{(t)}_{\boldsymbol{i}_{|t},\boldsymbol{i}_{|t'}} := A_{\boldsymbol{i}}\,.$$

The definition of the $\mathcal{H}$-Tucker format is quite involved, and as such we repeat [83, Definition 3.1, 3.4, and 3.6].

Definition 5.3.2. *A dimension tree or mode cluster tree T_d for dimension $d \in \mathbb{N}$ is a tree with root $\{1, \dots, d\}$ and depth l_{max} such that each node $t \in T_d$ is either*

1. *a leaf and singleton or*
2. *the union of two disjoint successors $S(t) = \{t_1, t_2\}$, i.e., $t = t_1 \cup t_2$.*

The level l of the tree is defined as the set of all nodes having a distance of exactly l to the root. We denote the level l of the tree by

$$T_d^l := \left\{ t \in T_d \colon \operatorname{level}(t) = l \right\}$$

The set of leaves of the tree is denoted by $\mathcal{L}(T_d)$.

Definition 5.3.3. *Let T_d be a dimension tree. The hierarchical rank $(r_t)_{t \in T_d}$ of a tensor $\boldsymbol{A} \in \mathbb{R}^{\mathcal{I}_{\boldsymbol{n}}}$ is defined by*

$$\forall t \in T_d \colon r_t := \operatorname{rank}(\boldsymbol{A}^{(t)}).$$

The set of all tensors of hierarchical rank (node-wise) at most $(r_t)_{t \in T_d}$ is denoted by

$$\mathcal{H}\text{-}Tucker((r_t)_{t \in T_d}) := \left\{ \boldsymbol{A} \in \mathbb{R}^{\mathcal{I}_{\boldsymbol{n}}} \colon \operatorname{rank}(\boldsymbol{A}^{(t)}) \leq r_t \right\}.$$

Definition 5.3.4. *Let T_d be a dimension tree and let $\boldsymbol{A} \in \mathcal{H}\text{-}Tucker((r_t)_{t \in T_d})$. Then $\boldsymbol{A}$ can be represented by transfer tensors $(\boldsymbol{V}_t)_{t \in T_d \setminus \mathcal{L}(T_d)}$ (for interior nodes) and mode frames $(\boldsymbol{U}_t)_{t \in \mathcal{L}(T_d)}$ (for leaves), where $\boldsymbol{V}_t \in \mathbb{R}^{r_t \times r_{t_1} \times r_{t_2}}$ for $S(t) = \{t_1, t_2\}$ and $\boldsymbol{U}_t \in \mathbb{R}^{\mathcal{I}_{\boldsymbol{n}|t} \times r_t}$. The complete representation of $\boldsymbol{A}$ is then given for all $t \in T_d \setminus \mathcal{L}(T_d)$ with sons $S(t) = \{t_1, t_2\}$ by the recursive relation*

$$\boldsymbol{U}_{t|\cdot,i} = \sum_{j_1=1}^{r_{t_1}} \sum_{j_2=1}^{r_{t_2}} \boldsymbol{V}_{t|i,j_1,j_2} \boldsymbol{U}_{t_1|\cdot,j_1} \otimes \boldsymbol{U}_{t_2|\cdot,j_2}, \quad i = 1, \dots, r_t,$$

with $\boldsymbol{A} = \boldsymbol{U}_{\{1,\dots,d\}}$. This decomposition contains $\mathcal{O}(d \max_k r_k n_k + (d-1) \max_k r_k^3)$ parameters.

A best approximation in the Frobenius norm of certain $\mathcal{H}$-Tucker rank always exists and a quasi-optimal $\mathcal{H}$-Tucker approximation with a certified maximal error can be computed [83, Theorem 3.22]. We shall denote the $\mathcal{H}$-Tucker leaves-to-root truncation operation of a tensor $\boldsymbol{A} \in \mathbb{R}^{\mathcal{I}_{\boldsymbol{n}}}$ up to a prescribed error ϵ as described in [83, Algorithm 2] by $\mathcal{T}_\epsilon$.

Theorem 5.3.1. *Let $\boldsymbol{A}_{best}$ be the best possible $\mathcal{H}$-Tucker approximation of the same $\mathcal{H}$-Tucker rank as $\mathcal{T}_\epsilon(\boldsymbol{A})$ in the Frobenius norm. We have then that*

$$\|\boldsymbol{A} - \mathcal{T}_\epsilon(\boldsymbol{A})\|_F \leq \epsilon \leq (2+\sqrt{2})\sqrt{d}\|\boldsymbol{A} - \boldsymbol{A}_{best}\|_F. \tag{5.15}$$

Note that the $\mathcal{H}$-Tucker truncation is quasi-optimal in the Frobenius norm. This does not necessarily imply closeness in the maximum norm, unless tensor (5.5) is, for example, generated by a smooth function [87].

5.3.3 A black-box low-rank tensor approximation algorithm

Our fuzzification strategy will rely on the black box approximation algorithm as described in [9]. The central idea of that algorithm is the application of adaptive cross approximation [13] (ACA) to the different matricizations $(\boldsymbol{B}^{(t)})_{t\in T_d}$ of a given tensor $\boldsymbol{B}$. These approximations are then used to construct an $\mathcal{H}$-Tucker approximation of increasing hierarchical rank until some convergence criterion is met.

For completeness we will recall the ACA algorithm from the above reference, together with a brief explanation of some of the underlying ideas. Also, the algorithm published in [9] leaves open some options for the implementer; hence, we will detail some of the choices we made in our implementation. First, we introduce the following notation. For a set of tensors/vectors $P := \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_r\} \subset \mathcal{I}_{\boldsymbol{n}}$, the notation $P_{\mathcal{J}}$ is used to denote the element-wise application of the entry selection, i.e., $P_{\mathcal{J}} := \{\boldsymbol{p}_{1|\mathcal{J}}, \ldots, \boldsymbol{p}_{r|\mathcal{J}}\}$.

The approximation that is iteratively constructed by the ACA algorithm is a pseudo-skeleton approximation [82] of type $\boldsymbol{X}^{(t)} := B^{(t)}_{\cdot,P^t_{t'}}(B^{(t)}_{P^t_t,P^t_{t'}})^{-1}B^{(t)}_{P^t_t,\cdot}$, where $P^t := \{\boldsymbol{p}^t_1, \ldots, \boldsymbol{p}^t_{r_t}\} \subset \mathcal{I}_{\boldsymbol{n}}$ is a set of so-called pivots. Important in this approximation is the *quality* of the submatrix $B^{(t)}_{P^t_t,P^t_{t'}}$. Generally, one will try to find a submatrix of maximal volume $\det(B^{(t)}_{P^t_t,P^t_{t'}})$. This is done heuristically by adding a new pivot to P^t that maximizes $|(\boldsymbol{B}^{(t)} - \boldsymbol{X}^{(t)})_{|i_t,i_{t'}}|$ over a matrix cross by an alternating search over i_t and $i_{t'}$.

The row and/or column size of a tensor matricization can be exceedingly large, so that a search over a complete matrix cross has to be avoided. In [9] this is resolved by doing the search for new pivots over tensor crosses instead. Moreover, for a correct $\mathcal{H}$-Tucker format, the sets of pivots $(P^t)_{t\in T_d}$ are ensured to be *nested*. The details are given in Algorithm 2.

Algorithm 2 produces an estimate ϵ_{est} of the error $\|\boldsymbol{B} - \boldsymbol{X}\|_\infty$ for each node $t \in T_d$. The details of how to use this error estimate in a stopping criterion

Algorithm 2 Greedy pivot search [9]

function NEWPIVOT(node t, sets of pivots $(P^t)_{t\in T_d}$, tensor $\boldsymbol{B}$)
  Let $\bar{t}$ be the brother and f the father of t (i.e., $S(f) = \{t, \bar{t}\}$)
  Choose a random pivot $\boldsymbol{i} \in \mathcal{I}_{\boldsymbol{n}}$ such that $i_{f'} \in P^f_{f'}$
  Virtually define current approximation to $\boldsymbol{B}^{(t)}$ by

$$\boldsymbol{X}^{(t)} \leftarrow B^{(t)}_{\cdot,P^t_{t'}} (B^{(t)}_{P^t_t,P^t_{t'}})^{-1} B^{(t)}_{P^t_t,\cdot}$$

  for $i_{\text{iter}} = 1, \ldots, n_{\text{iter}}$ **do**
    for $k \in f$ **do**
      Modify the kth entry of the index i by

$$i_k \leftarrow \operatorname*{arg\,max}_{i_k \in \mathcal{I}_{n_k}} |(\boldsymbol{B} - \boldsymbol{X})_{|i_1,\ldots,i_d}|$$

    end for
    Modify the remaining entries of the index $\boldsymbol{i}$ by

$$i_{f'} \leftarrow \operatorname*{arg\,max}_{i_{f'} \in P^f_{f'}} |(\boldsymbol{B} - \boldsymbol{X})_{|i_1,\ldots,i_d}|$$

  end for
  Define estimate of $\|\boldsymbol{B} - \boldsymbol{X}\|_\infty$ by

$$\epsilon_{\text{est}} \leftarrow |(\boldsymbol{B} - \boldsymbol{X})_{|i_1,\ldots,i_d}|$$

  return $\boldsymbol{i}, \epsilon_{\text{est}}$
end function

and how to cycle through the different nodes in the black box approximation algorithm are left open in [9]. In Algorithm 3, we fill in some of these blanks. The matricizations are constructed in a root-to-leaf order and a pivot is added to the set of pivots if the error estimate ϵ_{est} is larger than a fraction of the prescribed approximation error ϵ_{cross}.

Given that the final approximation $\boldsymbol{A}$ returned by the $\mathcal{H}$-Tucker cross approximation is of hierarchical rank $(r_t)_{t\in T_d}$, then [9, Lemma 7] shows that the algorithm is of

$$N_{\text{setup}} = \mathcal{O}\left(d r_{\max}^4 + l_{\max} r_{\max}^2 \sum_{k=1}^{d} n_k \right) \tag{5.16}$$

time complexity, where $r_{\max} := \max_{t\in T_d} r_t$, and needs

$$N_{\text{eval}} = \mathcal{O}\left(d r_{\max}^3 + l_{\max} r_{\max}^2 \sum_{k=1}^{d} n_k \right) \tag{5.17}$$

function evaluations. If the tree T_d is perfectly balanced, then its depth $l_{\max} = \lceil \log_2(d) \rceil$. The depth of a degenerate tree is $l_{\max} = d - 1$.

Algorithm 3 $\mathcal{H}$-Tucker cross approximation

function CROSSAPPR(tensor $\boldsymbol{B}$, accuracy ϵ_{cross})
  Choose a T_d for $\boldsymbol{B}$
  Initialize empty sets of pivots $(P^t)_{t\in T_d}$
  for $l = 1, \dots, l_{\max}$ **do**
    for $t \in T_d^l$ **do**
      repeat
        $\boldsymbol{i}, \epsilon_{\text{est}} \leftarrow$ NEWPIVOT(t, $(P^t)_{t\in T_d}$, $\boldsymbol{B}$)
        if $\epsilon_{\text{est}} \geq 0.1\epsilon_{\text{cross}}$ **then**
          $P^t \leftarrow P^t \cup \boldsymbol{i}$
        end if
      until $\epsilon_{\text{est}} \leq \epsilon_{\text{cross}}$
    end for
  end for
  Construct $\mathcal{H}$-Tucker approximation A:

$$\forall t \in \mathcal{L}(T_d)\colon \boldsymbol{U}_t \leftarrow B^{(t)}_{\cdot, P^t_{t'}}$$

$$\forall t \in T_d \setminus \mathcal{L}(T_d)\colon \boldsymbol{V}_{t|j_1,j_2,\cdot} \leftarrow$$

$$((B^{(t_1)}_{P^{t_1}_{t_1}, P^{t_1}_{t'_1}})^{-1}_{|j_1,\cdot} \otimes (B^{(t_2)}_{P^{t_2}_{t_2}, P^{t_2}_{t'_2}})^{-1}_{|j_2,\cdot})B^{(t)}_{(P^{t_1}_{t_1}, P^{t_2}_{t_2}), P^t_{t'}}$$

  return A
end function

Going back to the optimization approach as described in §5.3, we may now derive the following convergence result. If we assume for $n_1 = \cdots = n_d \to \infty$, that the true error of the low-rank approximations is bounded by ϵ_{trunc}, the error of finding the extremal elements in the slices of the tensor is bounded by ϵ_{opt}, and $r_{\max}$ is a bound for the maximum rank of all tensor approximations, then we may expect a fuzzification error

$$\epsilon_{\text{appr}} \leq \mathcal{O}\left(\frac{K d^{\frac{3}{2}} l_{\max} r_{\max}^2}{N_{\text{eval}} - K d r_{\max}^3}\right) + \epsilon_{\text{trunc}} + \epsilon_{\text{opt}}, \tag{5.18}$$

with K the constant from the complexity estimate (5.17). More simply, we can write

$$\epsilon_{\text{appr}} \leq \mathcal{O}\left(N_{\text{eval}}^{-1}\right) + \epsilon_{\text{trunc}} + \epsilon_{\text{opt}}. \tag{5.19}$$

Note however that the ACA algorithm can fail entirely [19]. A simple example is the tensor with only one non-zero entry. Even when the approximation is successful, the error estimate ϵ_{est} may not be a very good estimate of the actual

error [9, Lemma 14]. Whether smoothness or some particular structure of the underlying function may allow better estimates is an open problem.

5.3.4 Finding the maximal and the minimal entry in a low-rank tensor

Having constructed a low rank approximation to tensor $\boldsymbol{B}$, the next step in the fuzzification operation involves finding the maximal and minimal entries, as in (5.9). As described in [71], the search for the entry of maximal modulus in a tensor $\boldsymbol{A}$ is equivalent to the search for the eigenvalue of maximal modulus of the eigenvalue problem

$$\operatorname{diag}(\boldsymbol{A})\boldsymbol{v} = \lambda \boldsymbol{v}, \tag{5.20}$$

where $\operatorname{diag}(\boldsymbol{A})$ denotes the diagonalization of $\boldsymbol{A}$, i.e., the placement of all its entries on the diagonal of a matrix. Of course, solving this problem by traditional matrix-based methods should be avoided as the low-rank tensor structure will get lost.

In [71, Algorithm 1], a power iteration method is proposed with the essential modification that intermediate results are truncated back to a low-rank tensor. The power iteration method is not very efficient especially if the matrix is a diagonal matrix. In the $\mathcal{H}$-Tucker toolbox [127], we can find an implementation of the algorithm as described in Algorithm 4. Once the entry $\boldsymbol{i}_{\max}$ of maximal modulus is found, the other optimum can be found by a second application of Algorithm 4 to the tensor $\boldsymbol{A} - A_{\boldsymbol{i}_{\max}}$.

Algorithm 4 Maximal element in $\mathcal{H}$-Tucker tensor [71, 127]

function MAXELEM($\mathcal{H}$-Tucker tensor $\boldsymbol{A}$, truncation accuracy ϵ)
  while $\max_{t\in T_d}(r_t) > 1$ **do**
    $\boldsymbol{A} \leftarrow \mathcal{T}_\epsilon(\boldsymbol{A} \odot \boldsymbol{A})$ ▷ $\odot$ denotes the entrywise Hadamard product
    $\boldsymbol{A} \leftarrow \boldsymbol{A}/\|\boldsymbol{A}\|_{\mathrm{F}}$
  end while
  for $k = 1, \ldots, d$ **do**
    $i_k = \arg\max_j |\boldsymbol{V}_{\{k\}|j}|$
  end for
  return $\boldsymbol{i}$, $A_{\boldsymbol{i}}$
end function

5.4 Numerical experiments

We apply the low-rank approximation optimization scheme (which we will refer to as the HT algorithm) to three classical test functions which are of low-rank: the Rastrigin function, the Sinenvsin (or generalized Schaffer) function, and the Rosenbrock function [193]. As proposed in [139], we apply random translations and scalings of the standard search space of these functions to generate a more varied test set. Random rotations are not performed, as they increase the tensor-rank of the test functions.

As a reference, we also present the results of the GLCCLUSTER optimization algorithm (available in the TOMLAB Optimization Environment for Matlab [195]), and the Multilevel Coordinate Search (MCS) optimization algorithm [102]. These two solvers were found to be among the best algorithms in an extensive numerical study of derivative-free optimization solvers [179]. GLCCLUSTER is a hybrid solver which combines a global search by the Dividing Rectangles algorithm (DIRECT) [107] with a local search from clusters of promising points. DIRECT is a well-known algorithm of branching type. MCS also builds on ideas of the DIRECT algorithm, with alterations intended to deal with higher dimensional functions. Both solvers are applied with their default parameter settings. The local search option of the MCS algorithm is turned off, however, such that it can be run with a fixed amount of function evaluations.

For a meaningful comparison between the low-rank approximation fuzzification scheme and the two optimization algorithms, we take the following approach. Both GLCCLUSTER and MCS are transformed into a fuzzification strategy by independently applying these optimization algorithms to each of the global optimization problems in the fuzzification process. Because the sharing of information between the optimization problems on different α-cuts may result in a considerable reduction of the computation effort, we compute only one α-cut (i.e., the support). The gain from sharing information between a minimization and maximization in GLCCLUSTER or MCS is presumably rather limited. Note that the computation of only one α-cut is to the disadvantage of the HT algorithm in terms of needed number of function evaluations, because this algorithm does not need extra function evaluations to compute extra α-cuts.

The accuracy of the solutions returned by the algorithms is measured by comparison to the solution returned by the Interval Algorithm (INTERALG) (available in the software package OpenOpt [128]). This algorithm uses an interval analysis in a branch and bound framework to compute the exact solution to within a certain prescribed accuracy. Finally, Algorithm 4 is run

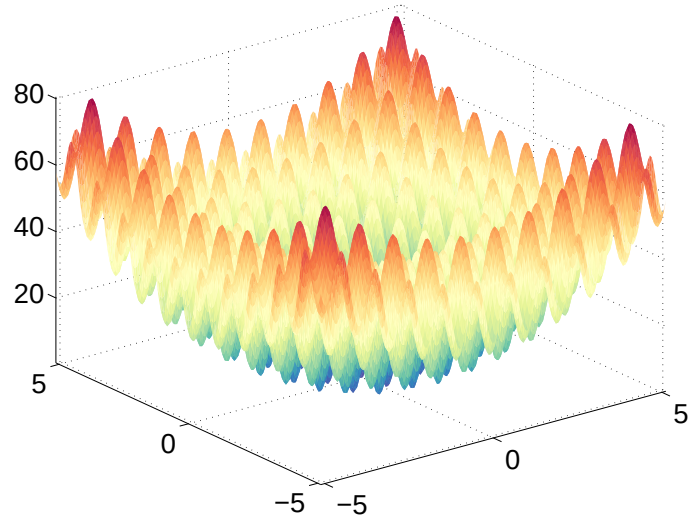


Figure 5.1: Plot of the Rastrigin function over the domain $[-5.12, 5.12]^2$.

with a fixed accuracy setting ($\epsilon = 10^{-14}$). With this setting, the error ϵ_{opt} was found to be well below the tolerance used in the computation of the INTERALG solution.

5.4.1 The Rastrigin function

The Rastrigin function is defined by

$$f\colon \boldsymbol{\xi} \mapsto 10d + \sum_{k=1}^{d} \left(\xi_k^2 - 10\cos(2\pi\xi_k)\right),$$

and has $\Omega := [-5.12, 5.12]^d$ as standard search space. The function is plotted in Fig. 5.1 for $d = 2$. The fuzzification algorithms are tested on ten fuzzy inputs constructed by the random translation and scaling within Ω of the symmetrical triangular fuzzy number $\widetilde{\boldsymbol{\xi}}$ with support $[\widetilde{\boldsymbol{\xi}}]_0 = [-5.12, 5.12] \times [-5.12/2, 5.12/2] \times \cdots \times [-5.12/d, 5.12/d]$ (algebraic decay). Using the INTERALG solution, the Rastrigin function is scaled for each randomly chosen $\widetilde{\boldsymbol{\xi}}$ such that its range over $[\widetilde{\boldsymbol{\xi}}]_0$ equals $[-1, 1]$. In that sense, absolute errors can be regarded as relative errors.

Table 5.1 lists the worst case—taken over ten randomly chosen fuzzy numbers—of an estimate of the error $\|\boldsymbol{A} - \boldsymbol{B}\|_\infty$ (found from the entries of $\boldsymbol{A} - \boldsymbol{B}$ closest

Table 5.1: The worst case—taken over ten randomly chosen fuzzy input numbers—of the estimated truncation error $\|\boldsymbol{A}-\boldsymbol{B}\|_\infty$, the maximum rank $r_{\max}$, the effective rank r_{eff}, and the number of function evaluations N_{eval} for the HT approximation of the Rastrigin function, $\epsilon_{\text{cross}} = 10^{-2}$, $n_{\text{iter}} = 2$, and several d and n_k.

	$n_k = 8$				$n_k = 32$			
d	$\|\boldsymbol{A}-\boldsymbol{B}\|_\infty$	$r_{\max}$	r_{eff}	N_{eval}	$\|\boldsymbol{A}-\boldsymbol{B}\|_\infty$	$r_{\max}$	r_{eff}	N_{eval}
5	$1.2\cdot10^{-15}$	2	2.0	2 512	$2.0\cdot10^{-15}$	2	2.0	8 224
10	$2.8\cdot10^{-15}$	2	2.0	7 440	$3.3\cdot10^{-15}$	2	2.0	24 624
15	$4.6\cdot10^{-15}$	2	2.0	13 040	$6.0\cdot10^{-15}$	2	2.0	43 424
20	$1.0\cdot10^{-14}$	2	2.0	19 536	$1.1\cdot10^{-14}$	2	2.0	65 424
	$n_k = 128$				$n_k = 512$			
d	$\|\boldsymbol{A}-\boldsymbol{B}\|_\infty$	$r_{\max}$	r_{eff}	N_{eval}	$\|\boldsymbol{A}-\boldsymbol{B}\|_\infty$	$r_{\max}$	r_{eff}	N_{eval}
5	$1.7\cdot10^{-15}$	2	2.0	31 072	$1.6\cdot10^{-15}$	2	2.0	122 464
10	$3.7\cdot10^{-15}$	2	2.0	93 360	$3.3\cdot10^{-15}$	2	2.0	368 304
15	$5.2\cdot10^{-15}$	2	2.0	164 960	$6.1\cdot10^{-15}$	2	2.0	651 104
20	$1.1\cdot10^{-14}$	2	2.0	248 976	$1.3\cdot10^{-14}$	2	2.0	983 184

to the INTERALG optima, the entries used in the HT algorithm, and an additional random selection), the maximum rank $r_{\max}$, the effective rank r_{eff}, and the number of function evaluations N_{eval}, for $\epsilon_{\text{cross}} = 10^{-2}$, $n_{\text{iter}} = 2$, and several d and n_k. The effective rank r_{eff} is defined in relation to the storage cost: $(d\, r_{\text{eff}} \max_k n_k + (d-1)r_{\text{eff}}^3)$. The size of the error $\|\boldsymbol{A}-\boldsymbol{B}\|_\infty$ shows that the HT algorithm is able to construct a very accurate low-rank approximation of $\boldsymbol{B}$ in all cases. The growth of the number of function evaluations with n_k and d is in agreement with estimate (5.17).

The exact α-cut $[f(\widetilde{\boldsymbol{\xi}})]_0$ is computed with the INTERALG algorithm up to an error tolerance smaller than $2\cdot10^{-3}$. This is used to compute an upper bound on ϵ_{appr}. Because of the observed bound $r_{\max} = 2$ on the rank, the very small observed error $\|\boldsymbol{A}-\boldsymbol{B}\|_\infty$, and an artificially small ϵ_{opt}, we expect an error $\epsilon_{\text{appr}} \leq \mathcal{O}(N_{\text{eval}}^{-1}) + \epsilon_{\text{trunc}} + \epsilon_{\text{opt}}$ (see (5.19)) for the HT algorithm, where $\epsilon_{\text{trunc}} \leq \|\boldsymbol{A}-\boldsymbol{B}\|_\infty$ and ϵ_{opt} are negligibly small in comparison to the accuracy of the INTERALG solution.

Fig. 5.2 shows the upper bound of the error ϵ_{appr} as a function of the number of function evaluations for ten randomly chosen fuzzy input numbers. The results of the HT algorithm are obtained with $\epsilon_{\text{cross}} = 10^{-2}$ and $n_k = 2, 4, 8, \ldots, 512$. The slope of the error curves of the HT algorithm seems to indicate an $\mathcal{O}(N_{\text{eval}}^{-2})$ behavior, which is faster than expected. This is likely due to a decreasing local Lipschitz constant in smaller and smaller neighborhoods of the optima. The

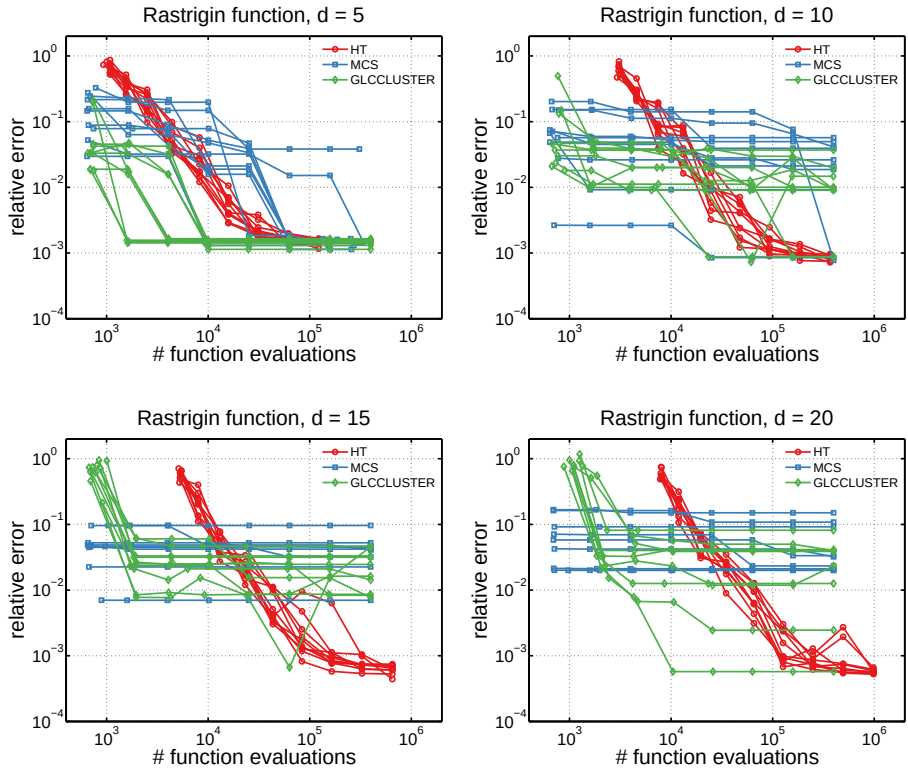


Figure 5.2: Upper bound of the error ϵ_{appr} for ten randomly chosen fuzzy input numbers.

slowing down of the convergence of the upper bound near 10^{-3} is due to the limited accuracy of the INTERALG solution, and not due to the error sources ϵ_{trunc} or ϵ_{opt}.

Fig 5.3 contains the timings as a function of N_{eval} for $d = 5$ and $d = 20$ (only one fuzzy input is shown in case of the HT algorithm). The computation time of the actual evaluation of the function is subtracted, such that the graphs only show the overhead generated by the algorithms. GLCCLUSTER starts off as the most efficient algorithm, but soon picks up an almost $\mathcal{O}(N_{\text{eval}}^3)$ complexity behavior. MCS has a linear $\mathcal{O}(N_{\text{eval}})$ complexity when $d = 5$. It reaches $\mathcal{O}(N_{\text{eval}}^2)$ when $d = 20$. The complexity of the HT algorithm grows very slowly. It is clear that the complexity of the low-rank tensor construction is dominated by the first term in (5.16) (note that the computation spent on the evaluation of the function is subtracted). The cost of Algorithm 4 initially grows as $\mathcal{O}(N_{\text{eval}})$

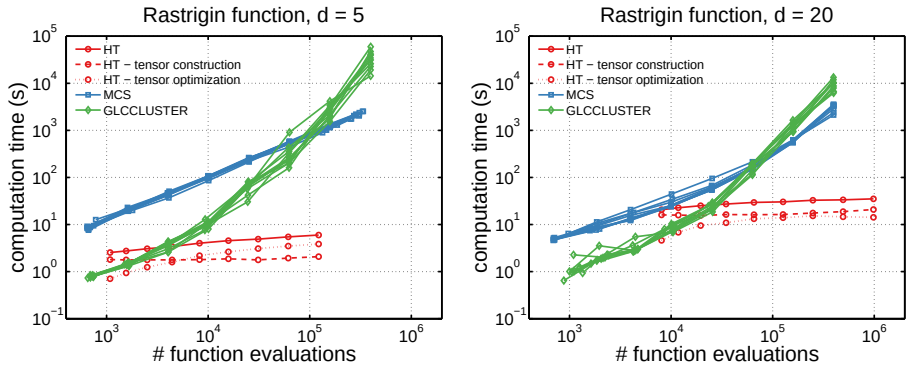


Figure 5.3: Computation time as a function of N_{eval} for ten randomly chosen fuzzy input numbers (only one in the case of the HT algorithm).

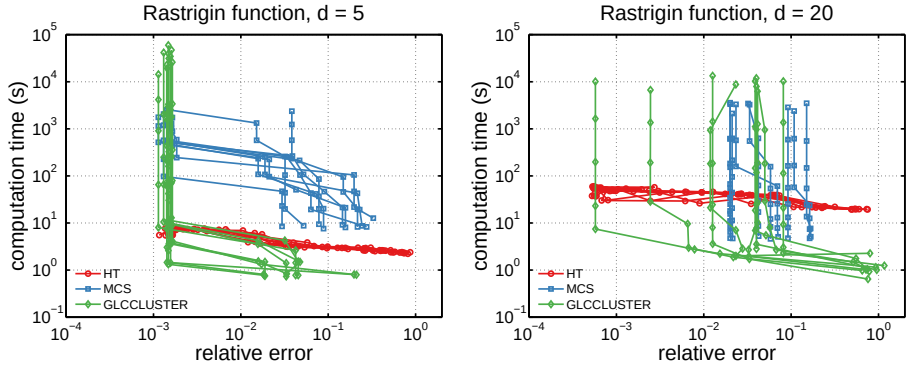


Figure 5.4: Computation time versus the upper bound of the error ϵ_{appr} for ten randomly chosen fuzzy input numbers.

but stabilizes for higher n_k when $d = 20$. It is our experience that the $\epsilon = 10^{-14}$ setting in Algorithm 4 is too accurate when n_k or d are small. How this setting influences the accuracy and the complexity of the algorithm is, however, out of the scope of this numerical study.

The question of which algorithm is best is not easily answered and depends on what criterion is used. For $d = 5$, it is clear that GLCCLUSTER is the better algorithm. For higher d, both GLCCLUSTER and MCS show a fast initial convergence but soon suffer from a severe drop in performance, possibly due to some kind of cluster problem or an ill balance of the global and the local search. The HT algorithm starts of with more function evaluations but soon

surpasses the other algorithms due to its fast convergence. Certainly when we plot the computation time versus the error (see Fig. 5.4), we can conclude that the HT algorithm is the best choice when a high accuracy is required. Another important observation that can be made is that the HT algorithm displays a very robust behavior with respect to the random variations in the test function. GLCCLUSTER and MCS hit a wall for $d = 20$, and are unable to improve the accuracy of their solution despite very high computation times.

5.4.2 The Sinenvsin function

The Sinenvsin function is a generalized Schaffer function and is defined by

$$f\colon \boldsymbol{\xi} \mapsto 0.5(d-1) + \sum_{k=1}^{d-1} \frac{\sin\left(\sqrt{\xi_k^2 + \xi_{k+1}^2}\right)^2 - 0.5}{\left(0.001(\xi_k^2 + \xi_{k+1}^2) + 1\right)^2}.$$

We choose $\Omega := [-10, 10]^d$ as the search space. Fig. 5.5 shows a plot for $d = 2$. The fuzzification algorithms are tested on ten fuzzy inputs constructed by the random translation and scaling within Ω of the symmetrical triangular fuzzy number $\widetilde{\boldsymbol{\xi}}$ with support $[\widetilde{\boldsymbol{\xi}}]_0 = [-10/2^0, 10/2^0] \times [-10/2^1, 10/2^1] \times \cdots \times [-10/2^{d-1}, 10/2^{d-1}]$ (exponential decay). Also here, the Sinenvsin function is scaled for each randomly chosen $\widetilde{\boldsymbol{\xi}}$ such that its range over $[\widetilde{\boldsymbol{\xi}}]_0$ equals $[-1, 1]$.

Table 5.2 lists the worst case—taken over ten randomly chosen fuzzy numbers—of an estimate of the error $\|\boldsymbol{A} - \boldsymbol{B}\|_\infty$, the maximum rank $r_{\max}$, the effective rank r_{eff}, and the number of function evaluations N_{eval} for $n_k = 128$, $n_{\text{iter}} = 2$, several d and ϵ_{cross}. The size of the error $\|\boldsymbol{A} - \boldsymbol{B}\|_\infty$ shows that the HT algorithm is able to construct a low-rank approximation of $\boldsymbol{B}$ with the desired accuracy ϵ_{cross} reasonably well except for $\epsilon_{\text{cross}} = 10^{-8}$. This seems to indicate an instability in the HT approximation algorithm. The growth of the number of function evaluations with d is in correspondence with estimate (5.17).

The exact α-cut $[f(\widetilde{\boldsymbol{\xi}})]_0$ is computed with the INTERALG algorithm up to an error tolerance smaller than 10^{-4}. Fig. 5.6 shows the upper bound of the error ϵ_{appr} as a function of the number of function evaluations for ten randomly chosen fuzzy numbers. The results of the HT algorithm are obtained with $\epsilon_{\text{cross}} = 10^{-2}$ and $n_k = 2, 4, 8, \ldots, 512$. The slope of the error curves of the HT algorithm indicates an $\mathcal{O}(N_{\text{eval}}^{-2})$ behavior, which is again faster than expected.

Fig. 5.7 contains the timings for $d = 5$ and $d = 20$ for ten randomly chosen fuzzy input numbers (only one in the case of the HT algorithm). GLCCLUSTER starts off as the most efficient algorithm, but soon picks up an $\mathcal{O}(N_{\text{eval}}^2)$

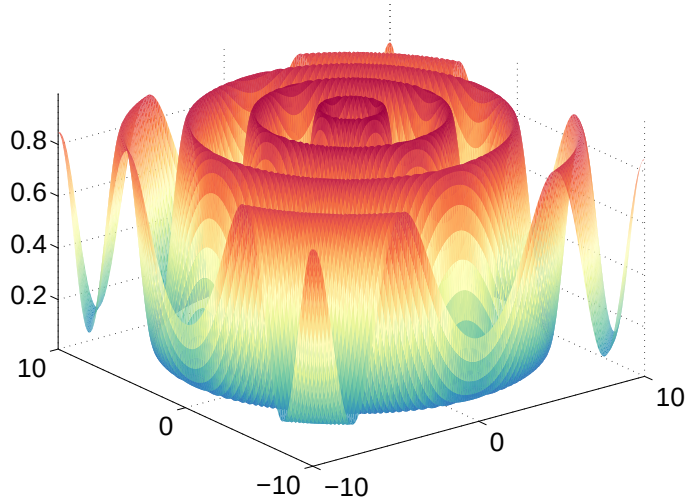


Figure 5.5: Plot of the Sinenvsin function over the domain $[-10, 10]^2$.

Table 5.2: The worst case—taken over ten randomly chosen fuzzy input numbers—of the estimated truncation error $\|\boldsymbol{A} - \boldsymbol{B}\|_\infty$, the maximum rank $r_{\max}$, the effective rank r_{eff}, the effective rank r_{eff}, and the number of function evaluations N_{eval} for the HT approximation of the Sinenvsin function, $n_k = 128$, $n_{\text{iter}} = 2$, and several d and ϵ_{cross}.

	$\epsilon_{\text{cross}} = 10^{-2}$				$\epsilon_{\text{cross}} = 10^{-4}$			
d	$\|\boldsymbol{A} - \boldsymbol{B}\|_\infty$	$r_{\max}$	r_{eff}	N_{eval}	$\|\boldsymbol{A} - \boldsymbol{B}\|_\infty$	$r_{\max}$	r_{eff}	N_{eval}
5	$1.5 \cdot 10^{-3}$	8	4.8	97 600	$1.2 \cdot 10^{-5}$	9	5.8	132 081
10	$3.1 \cdot 10^{-3}$	9	3.6	191 241	$7.1 \cdot 10^{-4}$	10	4.1	276 058
15	$4.8 \cdot 10^{-3}$	9	2.7	251 862	$1.6 \cdot 10^{-5}$	11	3.3	349 632
20	$3.6 \cdot 10^{-3}$	9	2.4	274 777	$1.8 \cdot 10^{-5}$	9	2.8	379 304
	$\epsilon_{\text{cross}} = 10^{-6}$				$\epsilon_{\text{cross}} = 10^{-8}$			
d	$\|\boldsymbol{A} - \boldsymbol{B}\|_\infty$	$r_{\max}$	r_{eff}	N_{eval}	$\|\boldsymbol{A} - \boldsymbol{B}\|_\infty$	$r_{\max}$	r_{eff}	N_{eval}
5	$9.1 \cdot 10^{-8}$	10	6.3	157 412	$1.6 \cdot 10^{-5}$	10	7.1	194 992
10	$9.2 \cdot 10^{-7}$	11	4.8	344 425	$1.5 \cdot 10^{-5}$	12	5.2	387 743
15	$8.6 \cdot 10^{-6}$	12	4.0	430 638	$5.9 \cdot 10^{-5}$	13	4.4	520 638
20	$1.9 \cdot 10^{-7}$	10	3.3	483 365	$3.8 \cdot 10^{-5}$	11	3.8	572 790

complexity behavior. Also here, MCS has a linear $\mathcal{O}(N_{\text{eval}})$ complexity when $d = 5$. It reaches approximately $\mathcal{O}(N_{\text{eval}}^2)$ when $d = 20$.

When only taking into account the number of function evaluations (Fig. 5.6),

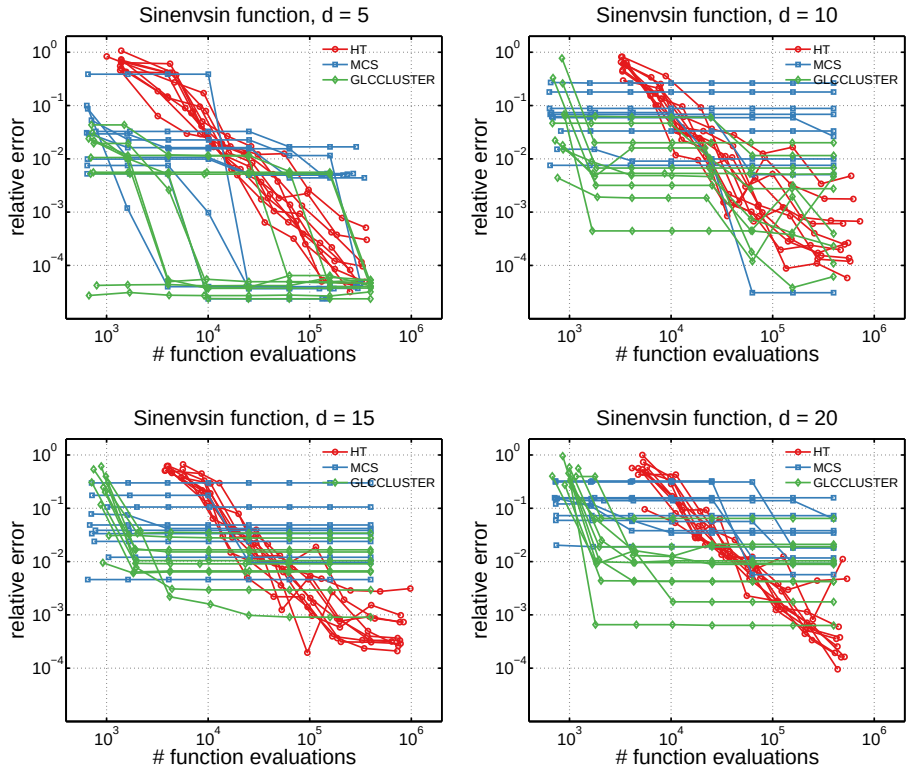


Figure 5.6: Upper bound of the error ϵ_{appr} for ten randomly chosen fuzzy input numbers.

GLCCLUSTER is again clearly the best algorithm for $d = 5$. For higher d, both GLCCLUSTER and MCS show a fast initial convergence but then suffer from a severe drop in performance. The HT algorithm starts off with more function evaluations but soon surpasses the other algorithms due to its fast convergence. Certainly when we plot the computation time versus the error (see Fig. 5.8), we can conclude that the HT algorithm seems again to be the best choice when a high accuracy is required. Also the robust behavior of the HT algorithm can be observed again.

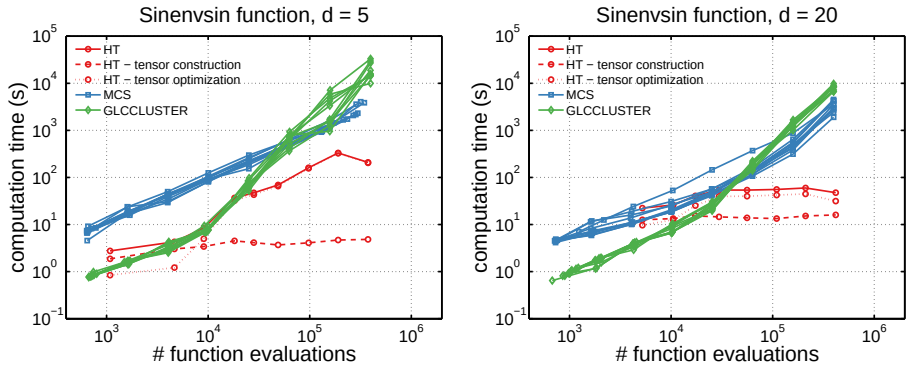


Figure 5.7: Computation time as a function of N_{eval} for ten randomly chosen fuzzy input numbers (only one in the case of the HT algorithm).

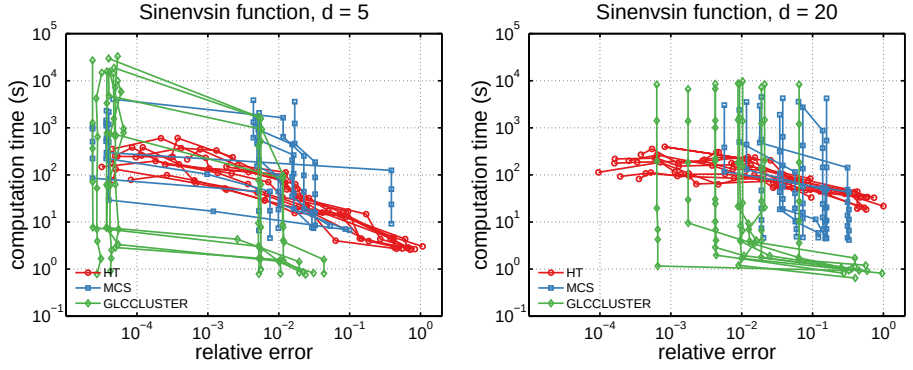


Figure 5.8: Computation time versus the upper bound of the error ϵ_{appr} for ten randomly chosen fuzzy input numbers.

5.4.3 The Rosenbrock function

As a final example, we choose the following higher dimensional generalization of the Rosenbrock function:

$$f\colon \boldsymbol{\xi} \mapsto \sum_{k=1}^{d-1} (\xi_k - 1)^2 + 100(\xi_{k+1} - \xi_k^2)^2 \,.$$

Compared to the earlier 2 examples, this function is much less oscillatory, so we expect the classical methods to perform quite well. We choose $\Omega := [0.5, 1.5]^d$ as the search space. Fig. 5.9 shows a plot for $d = 2$. The fuzzification algorithms

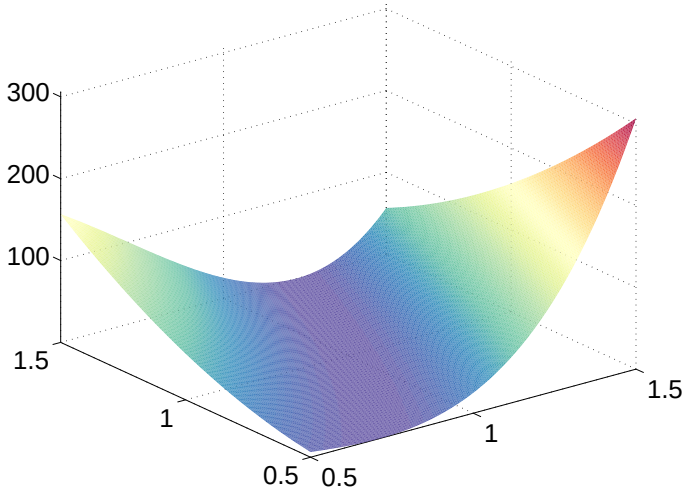


Figure 5.9: Plot of the Rosenbrock function over the domain $[0.5, 1.5]^2$.

are tested on ten fuzzy inputs constructed by the random translation and scaling within Ω of the symmetrical triangular fuzzy number $\widetilde{\boldsymbol{\xi}}$ with support $[\widetilde{\boldsymbol{\xi}}]_0 = [1-0.5, 1+0.5] \times [1-0.5/2, 1+0.5/2] \times \cdots \times [1-0.5/d, 1+0.5/d]$ (algebraic decay). The Rosenbrock function are scaled for each randomly chosen $\widetilde{\boldsymbol{\xi}}$ such that its range over $[\widetilde{\boldsymbol{\xi}}]_0$ equals $[-1, 1]$.

Table 5.3 lists the worst case—taken over ten randomly chosen fuzzy numbers—of an estimate of the error $\|\boldsymbol{A} - \boldsymbol{B}\|_\infty$, the maximum rank $r_{\max}$, the effective rank r_{eff}, and the number of function evaluations N_{eval} for $n_k = 128$, $n_{\text{iter}} = 2$, several d and ϵ_{cross}. These results show that the low-rank approximation by the HT algorithm is able to construct a low-rank approximation of B with the desired accuracy except for $\epsilon_{\text{cross}} = 10^{-8}$. This indicates again some instability in the HT approximation algorithm.

The exact α-cut $[f(\widetilde{\boldsymbol{\xi}})]_0$ is computed with the INTERALG algorithm up to an error tolerance smaller than 10^{-5}. Fig. 5.10 shows the upper bound of the error ϵ_{appr} as a function of the number of function evaluations for ten randomly chosen fuzzy numbers. The results of the HT algorithm are obtained with $\epsilon_{\text{cross}} = 10^{-6}$ and $n_k = 2, 4, 8, \ldots, 256$. The slope of the error curves of the HT algorithm indicates an $\mathcal{O}(N_{\text{eval}}^{-3})$ behavior.

Fig. 5.11 contains the timings for $d = 5$ and $d = 20$ for ten randomly chosen fuzzy input numbers (only one in the case of the HT algorithm). We can make very similar observations as with the Rastrigin and the Sinenvsin function.

Table 5.3: The worst case—taken over ten randomly chosen fuzzy numbers—of the estimated truncation error $\|\boldsymbol{A}-\boldsymbol{B}\|_\infty$, the maximum rank $r_{\max}$, the effective rank r_{eff}, the effective rank r_{eff}, and the number of function evaluations N_{eval} for the HT approximation of the Rosenbrock function, $n_k = 128$, $n_{\text{iter}} = 2$, and several d and ϵ_{cross}.

	$\epsilon_{\text{cross}} = 10^{-2}$				$\epsilon_{\text{cross}} = 10^{-4}$			
d	$\|\boldsymbol{A}-\boldsymbol{B}\|_\infty$	$r_{\max}$	r_{eff}	N_{eval}	$\|\boldsymbol{A}-\boldsymbol{B}\|_\infty$	$r_{\max}$	r_{eff}	N_{eval}
5	$1.9 \cdot 10^{-3}$	4	3.2	49 942	$1.2 \cdot 10^{-5}$	4	3.3	59 114
10	$1.3 \cdot 10^{-3}$	4	2.5	102 576	$1.2 \cdot 10^{-5}$	4	2.9	150 145
15	$1.8 \cdot 10^{-3}$	4	2.1	170 480	$1.6 \cdot 10^{-5}$	4	2.5	217 144
20	$1.9 \cdot 10^{-3}$	4	1.7	186 732	$2.4 \cdot 10^{-5}$	4	2.2	284 119
	$\epsilon_{\text{cross}} = 10^{-6}$				$\epsilon_{\text{cross}} = 10^{-8}$			
d	$\|\boldsymbol{A}-\boldsymbol{B}\|_\infty$	$r_{\max}$	r_{eff}	N_{eval}	$\|\boldsymbol{A}-\boldsymbol{B}\|_\infty$	$r_{\max}$	r_{eff}	N_{eval}
5	$5.5 \cdot 10^{-8}$	4	3.5	61 968	$8.8 \cdot 10^{-7}$	4	3.5	61 968
10	$1.2 \cdot 10^{-7}$	4	3.2	173 451	$5.6 \cdot 10^{-7}$	4	3.5	187 139
15	$6.6 \cdot 10^{-7}$	4	2.8	268 538	$1.8 \cdot 10^{-6}$	4	3.1	289 538
20	$1.4 \cdot 10^{-7}$	4	2.6	344 153	$9.4 \cdot 10^{-6}$	4	2.8	363 107

As with the Sinenvsin function, the HT algorithm is clearly dominated by Algorithm 4.

It seems that MCS is the best algorithm overall, although there are a few fuzzy input numbers for $d = 20$ where the convergence is very slow. The performance of GLCCLUSTER is rather poor, especially for $d = 20$. The HT algorithm does not really compete with MCS except for $d = 20$, where its robust behavior can be an advantage if reliability of the outcome is important. This is certainly true when taking into account the computation time (see Fig. 5.12). Then, the HT algorithm proves to be an attractive choice.

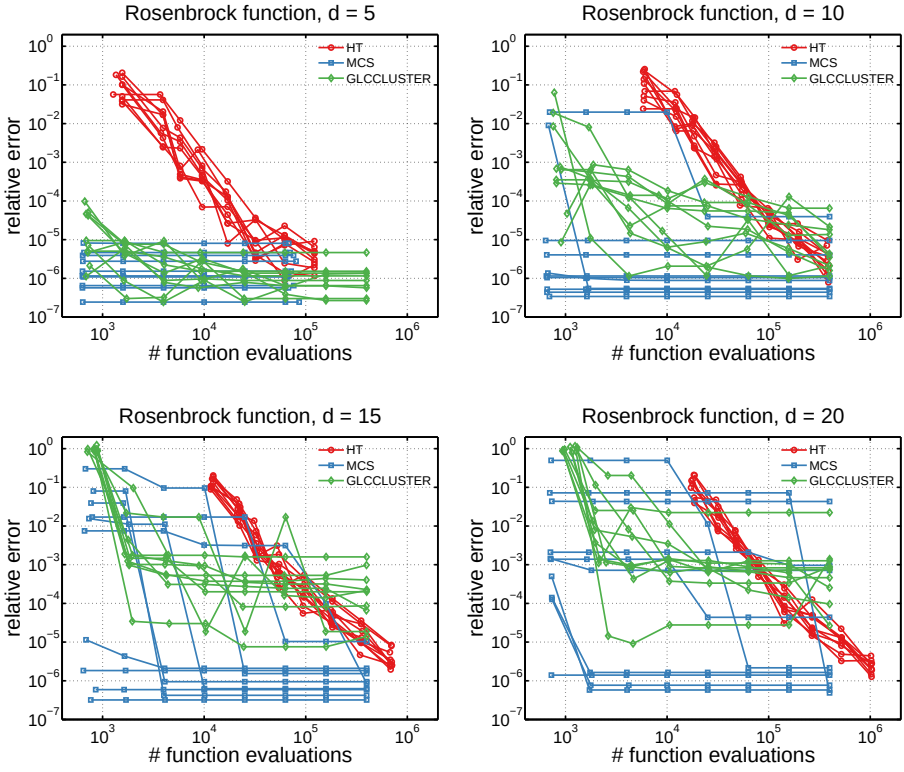


Figure 5.10: Upper bound of the error ϵ_{appr} for ten randomly chosen fuzzy input numbers.

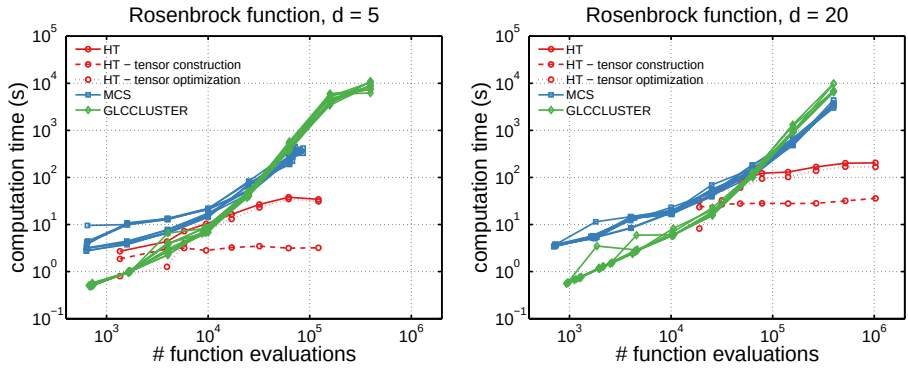


Figure 5.11: Computation time as a function of N_{eval} for ten randomly chosen fuzzy input numbers (only one in the case of the HT algorithm).

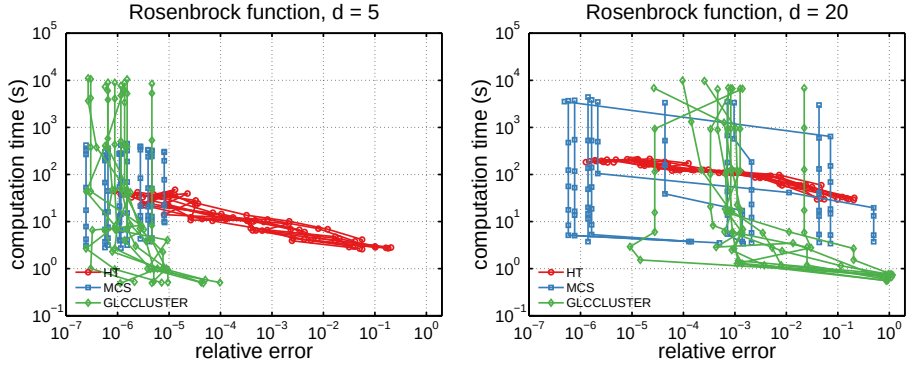


Figure 5.12: Computation time versus the upper bound of the error ϵ_{appr} for ten randomly chosen fuzzy input numbers.

5.5 Conclusions

This chapter presented an algorithm for the fuzzification of a continuous function based on a low-rank tensor approximation of the function on a grid. This algorithm consists of two stages: first, a low-rank tensor approximation and then the search for the minimal and maximal entries in the low-rank tensor. The low-rank tensor approximation of the function on a grid can be constructed in several ways. In this chapter, we used the tensor cross approximation algorithm as developed by Ballani et al. [9]. It yields a tensor

approximation in the hierarchical Tucker format. The actual optimization (the search for the minimal and maximal entries in parts of the low-rank tensor) was done with the algorithm as found in the $\mathcal{H}$-Tucker toolbox [127]. Exchange of information between the different optimization problems in the α-cut approach is established by the shared construction of the low-rank tensor approximation.

Despite the limited scope of the numerical study that we presented, we can conclude that there is a large potential in the use of low-rank tensor methods in optimization. Whereas advanced global optimization algorithms such as MCS and GLCCLUSTER have an increasing difficulty to deal with higher dimensional search spaces, the low-rank approximation approach seems to effectively break the curse of dimensionality for the functions we tested. Despite the promising results, theoretical results of convergence, complexity and stability of both the low-rank tensor approximation algorithm and the algorithm which finds the extremal elements in a low-rank tensor are, however, still largely missing.

Chapter 6

Recycling strategies for the solution of sequences of similar linear systems

In the construction of a response surface approximation by a collocation approach or the direct computation of quantities of interest of a fuzzy partial differential equation by the α-cut approach, many similar systems have to be solved. These systems and corresponding solutions can be such that a considerable reduction in computational cost can be achieved if some information is recycled from one system to the next. This chapter will discuss a few of these approaches. The results of this chapter have been partly published in [37].

6.1 Introduction

Our motivating example problem is again the fuzzy diffusion equation on a d_Ω-dimensional Lipschitz domain $\Omega \subset \mathbb{R}^{d_\Omega}$:

$$-\nabla \cdot (\widetilde{a}\, \nabla \widetilde{u}) = f, \qquad \widetilde{u}|_{\partial\Omega} = 0, \tag{6.1}$$

with $f \in H^{-1}(\Omega)$ the deterministic source term, $\widetilde{a} \in \mathcal{F}(L^\infty(\Omega))$ the fuzzy input field, and $\widetilde{u} \in \mathcal{F}(H_0^1(\Omega))$ the unknown fuzzy solution field. We consider a fuzzy

input field that is (or has been approximated by) an expansion of the form, i.e.,

$$\widetilde{a} = a_0 + \sigma \sum_{k=1}^{d_\Xi} a_k \widetilde{\xi}_k \tag{6.2}$$

with $a_k \in L^\infty(\Omega)$, $k = 0, \ldots, d_\Xi$, and $[\widetilde{\boldsymbol{\xi}}]_0 = [-1, 1]^{d_\Xi}$. In order to ensure that the problem is well-posed, we assume that there exist real numbers $a_{\min}$ and $a_{\max}$ for which

$$0 < a_{\min} \leq a(\boldsymbol{x}, \boldsymbol{\xi}) \leq a_{\max} < \infty \tag{6.3}$$

almost everywhere in Ω and for all $\boldsymbol{\xi} \in [\widetilde{\boldsymbol{\xi}}]_0$.

As discussed in §3.3.1, a common approach to compute the solution of this fuzzy equation is to first construct a response surface approximation. If this response surface is of interpolating type in the parameter domain (see, e.g., §3.3.6), then a linear system of the form $\boldsymbol{A}(\boldsymbol{\xi})\boldsymbol{u}(\boldsymbol{\xi}) = \boldsymbol{f}(\boldsymbol{\xi})$ will have to be solved for each interpolation node. The linear systems which result from a discretization of the parameterized equation corresponding to (6.1) are of such form that they allow a very efficient solution by a Conjugate Gradient (CG) solver with a preconditioner of multigrid type.

Many of the approaches that can be found in the literature to reduce the computation time of solving long sequences of similar linear systems are based on the premise that an iterative solver is used. Arguably the simplest approach that can be found in the literature is the reuse of a previous solution or a linear combination of previous solutions as initial guess for the next system in the sequence [73]. Another straightforward approach is the simple reuse of the preconditioner constructed for solving one reference system to solve all other systems. In the structural engineering literature, this approach is sometimes referred to as the Combined Approximations algorithm [72, 117, 118]. In the context of the collocation approach to solve stochastic differential equations, this preconditioner is better known as the mean-based preconditioner (where the reference point is equal to the mean) [69, 174, 199]. Finally, a more complicated approach that attracted considerable attention is that of recycling Krylov subspace vectors from one system to the next [28, 116, 168, 189, 211].

In [37], we proposed a method in which the prolongation and restriction operators of the multigrid preconditioner of one reference system are recycled to all other systems. In an algebraic multigrid (AMG) procedure, the construction of these intergrid transfer operators can amount to a significant portion of the total computation time. By recycling them, a preconditioner can be constructed that is better tuned to each individual system than the "mean-based" preconditioner. The numerical results that we will present show that

this preconditioner performs almost as well as the AMG preconditioner which is constructed for each system individually, while being much cheaper to construct. For a more elaborate and theoretical discussion of this AMG preconditioner with recycled setup, we refer to [81].

Apart from the AMG precondioner with recycled setup, we will make use of two other recycling strategies. First, we adopt the approach of reusing a previous solution as initial guess for the next system to solve. In that regard, it is important to solve the systems in a particular order such that the previous solutions provide a reasonable initial guess. We will discuss such an ordering strategy. Second, the construction of the matrix $\boldsymbol{A}(\boldsymbol{\xi})$ from scratch by running the FEM software for each new $\boldsymbol{\xi}$ can be rather expensive. The specific form of $\widetilde{a}(\boldsymbol{x})$ however allows to construct $\boldsymbol{A}(\boldsymbol{\xi})$ as a linear combination of precomputed matrices $\boldsymbol{A}_0, \ldots, \boldsymbol{A}_{d_\Xi}$ (see equation (3.96)). This obvious step can significantly reduce the computational effort.

6.2 Reuse of an earlier solution as an initial guess

The all-zeros vector as an initial guess is a standard choice when using an iterative method to solve the system $\boldsymbol{A}(\boldsymbol{\xi})\boldsymbol{u}(\boldsymbol{\xi}) = \boldsymbol{f}(\boldsymbol{\xi})$. There are however several options to construct a better initial guess when this system has to be solved repeatedly for many different values of $\boldsymbol{\xi}$. By construction of a response surface approximation of $\boldsymbol{\xi} \mapsto \boldsymbol{u}(\boldsymbol{\xi})$ based on the solutions of previously solved systems, one can make a more informed guess of the solution of the next system. This response surface can be part of the context in which this sequence of systems is generated in such a way that no extra cost is involved in the construction of such a response surface. Such a context could be the construction of an adaptive sparse grid interpolating polynomial surface [76, 119, 141].

The cost of the construction and evaluation of the response surface should be weighed against the potential gain in efficiency in solving the linear systems. Sometimes it may be more beneficial—or just simpler to implement—to reuse a previous solution as an initial guess. Gordon et al. [81] use a strategy in which the systems are ordered in a greedy fashion by selecting the next system to solve as the one that is closest to the last solved system. The last solution is then reused as an initial guess. There, the systems originate from the discretization of a mixed formulation of the steady-state diffusion equation with a stochastic diffusion coefficient of the form equivalent to (6.2). Closeness of two systems at $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ is expressed by the weighted 1-norm $\sum_{k=1}^{d_\Xi} \|a_k\|_{L^2(\Omega)} |\boldsymbol{\xi}_{1|k} - \boldsymbol{\xi}_{2|k}|$.

The approach that we will use is based on the assumption that all solutions of previously solved systems (or a subset thereof) are stored. In that case, we can improve on the approach described by Gordon et al. [81]. We select the next system to solve as the one that is closest to any of the previously solved and stored systems, and correspondingly use that system's solution as an initial guess. We will also use a somewhat more general distance measure which does not assume any knowledge of the structure of the systems. Closeness of two systems at $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ will simply be expressed by the Euclidean norm $\|\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2\|_2$.

6.3 Reuse of prolongation and restriction operators in algebraic multigrid

For the reader's convenience, we repeat the short explanation of multigrid here that was given in Section 4.1.2. Multigrid methods are among the fastest techniques for solving the sparse linear system which results from a finite element discretization of the parameterized equation corresponding to (6.1). Multigrid is an iterative method which combines a smoothing operation and a coarse-grid correction in a recursive manner; see Algorithm 5. A prolongation operator $\boldsymbol{P}_{l-1,l}$ and a restriction operator $\boldsymbol{R}_{l,l-1}$ transfer corrections and residuals back and forth between different levels $l = l_{\max}, l_{\max} - 1, \ldots, 0$. On every level l, there is a corresponding operator $\boldsymbol{L}_l$. The smoothing operation $\boldsymbol{S}_l$ is typically a one-level method based on a matrix splitting (see (4.8)).

The intergrid transfer operators $\boldsymbol{P}_{l-1,l}$ and $\boldsymbol{R}_{l,l-1}$, and the operators $\boldsymbol{L}_l$ are constructed during setup of the multigrid method. In algebraic multigrid, the operators $\boldsymbol{P}_{l-1,l}$ and $\boldsymbol{R}_{l,l-1}$ are constructed from a repeated algebraic coarsening of the operator $\boldsymbol{A}$. The operators $\boldsymbol{L}_l$ are constructed based on the Galerkin principle:

$$\boldsymbol{L}_{l-1} = \boldsymbol{P}_{l-1,l} \boldsymbol{L}_l \boldsymbol{R}_{l,l-1} \tag{6.4}$$

with $\boldsymbol{L}_{l_{\max}} := \boldsymbol{A}$.

The algebraic coarsening step is expensive and, as such, the setup of the multigrid method can amount to a significant portion of the total computation time. Therefore, in the case of having to solve many similar linear systems, one can consider reusing the restriction and prolongation operators $\boldsymbol{P}_{l-1,l}^{\boldsymbol{\xi}_{\text{ref}}}$ and $\boldsymbol{R}_{l,l-1}^{\boldsymbol{\xi}_{\text{ref}}}$ constructed for some reference system $\boldsymbol{A}(\boldsymbol{\xi}_{\text{ref}})\boldsymbol{u}(\boldsymbol{\xi}_{\text{ref}}) = \boldsymbol{f}(\boldsymbol{\xi}_{\text{ref}})$. At a new point $\boldsymbol{\xi}$, the hierarchy is then constructed using cheap sparse matrix multiplications:

$$\boldsymbol{L}_{l-1} = \boldsymbol{P}_{l-1,l}^{\boldsymbol{\xi}_{\text{ref}}} \boldsymbol{L}_l \boldsymbol{R}_{l,l-1}^{\boldsymbol{\xi}_{\text{ref}}} \tag{6.5}$$

Algorithm 5 Multigrid iteration

function MULTIGRID(approximation $\boldsymbol{u}_l$, right hand side $\boldsymbol{f}_l$, level l)
  if $l = 0$ **then**
    $\boldsymbol{u}_0 \leftarrow \boldsymbol{L}_0^{-1} \boldsymbol{f}_0$
  else
    Pre-smoothing: apply smoother ν_1 times, i.e., $\boldsymbol{u}_l \leftarrow \boldsymbol{S}_l^{\nu_1}(\boldsymbol{u}_l, \boldsymbol{L}_l, \boldsymbol{f}_l)$
    Restrict residual to coarser level: $\boldsymbol{f}_{l-1} \leftarrow \boldsymbol{R}_{l,l-1}(\boldsymbol{f}_l - \boldsymbol{L}_l \boldsymbol{u}_l)$
    Coarse grid correction:
    $\boldsymbol{e}_{l-1} \leftarrow 0$
    for $i = 1, \dots, \gamma$ **do**
      $\boldsymbol{e}_{l-1} \leftarrow$ MULTIGRID$(\boldsymbol{e}_{l-1}, \boldsymbol{f}_{l-1}, l-1)$
    end for
    Prolongate correction and update approximation:

$$\boldsymbol{u}_l \leftarrow \boldsymbol{u}_l + \boldsymbol{P}_{l-1,l} \boldsymbol{e}_{l-1}$$

    Post-smoothing: apply smoother ν_2 times, i.e.,

$$\boldsymbol{u}_l \leftarrow \boldsymbol{S}_l^{\nu_2}(\boldsymbol{u}_l, \boldsymbol{L}_l, \boldsymbol{f}_l)$$

  end if
  return $\boldsymbol{u}_l$
end function

with $\boldsymbol{L}_{l_{\max}} := \boldsymbol{A}(\boldsymbol{\xi})$.

6.4 Numerical experiments

As an example, we consider the steady-state diffusion equation (6.1) on a square domain $\Omega := (0,1)^2$ with $f(\boldsymbol{x}) = 1$ and a piecewise constant diffusion coefficient

$$\widetilde{a}(\boldsymbol{x}) = \begin{cases} 0.5 + \sigma \widetilde{\xi}_k & \text{if } \boldsymbol{x} \in \mathcal{D}_k,\ k = 1, \dots, d_\Xi, \\ 1 & \text{if } \boldsymbol{x} \in \Omega \setminus \bigcup_{k=1}^{d_\Xi} \mathcal{D}_k, \end{cases} \tag{6.6}$$

where $\mathcal{D}_1, \dots, \mathcal{D}_{d_\Xi}$ are evenly distributed mutually disjoint discs; see Fig. 6.1 for an illustration of the geometry with $d_\Xi = 9$. The fuzzy numbers $\widetilde{\xi}_1, \dots, \widetilde{\xi}_{d_\Xi}$ are assumed to be noninteractive symmetrical triangular fuzzy numbers with support $[-1, 1]$. The scaling factor σ will be chosen such that all $a \in [\widetilde{a}]_0$ are strictly positive.

We construct a response surface based on a sparse grid interpolation, i.e., we compute $I_{\mathcal{J}^{\mathrm{SM}}_{\boldsymbol{w},p}} u^{\mathrm{r}}_{n_\phi}$ with $\mathcal{J}^{\mathrm{SM}}_{\boldsymbol{w},p}$ as defined in (3.83). We will use a weight $\boldsymbol{w} = (1, \dots, 1)$ (isotropic grid). The integer p will be called the level of the sparse grid. The sparse grid interpolation incurs the need to solve many similar

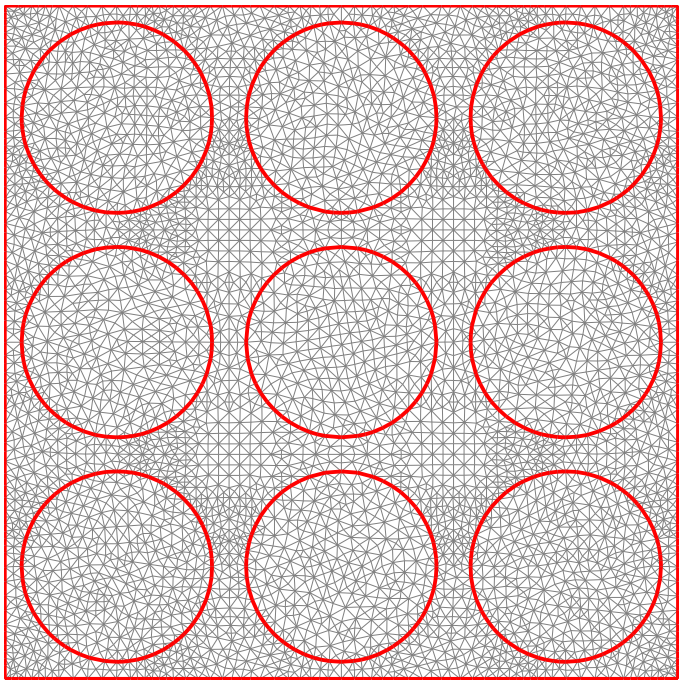


Figure 6.1: Geometry and mesh with $n_\phi = 4\,038$ degrees of freedom for $d_\Xi = 9$.

linear systems, where the linear systems originate from a discretization of the parametric PDE corresponding to (6.1). The spatial discretization $u^{\mathrm{r}}_{n_\phi}$ (see equation (3.95)) will be done using triangular finite elements with piecewise linear basis functions; see Fig. 6.1 for a mesh with $n_\phi = 4038$ degrees of freedom. Because of the strict positivity of the diffusion coefficient over the whole support, all linear systems are positive definite. This allows us to use a CG solver (the systems are also symmetric). For the preconditioner, we use one multigrid V(1,1)-cycle (i.e., $\gamma = 1$, $\nu_1 = \nu_2 = 1$) with a Ruge–Stüben hierarchy [192] and a symmetric Gauss–Seidel smoother.

We study the effect of different parameters on the timings of different recycling strategies. Concerning the reuse of an earlier solution as initial guess, we employ four recycling strategies:

- **zeros:** The all-zeros vector is used as initial guess.
- **random:** Random ordering of the systems, and reuse of the solution of the last solved system as initial guess.

- **Gordon et al.:** Ordering of the systems as described in [81], and reuse of the solution of the last solved system as initial guess. Closeness of two systems is, however, expressed by the Euclidean distance.
- **nearest:** Ordering of systems and reuse of solution as initial guess as described in §6.2.

For the construction of the multigrid preconditioner, we test three recycling strategies:

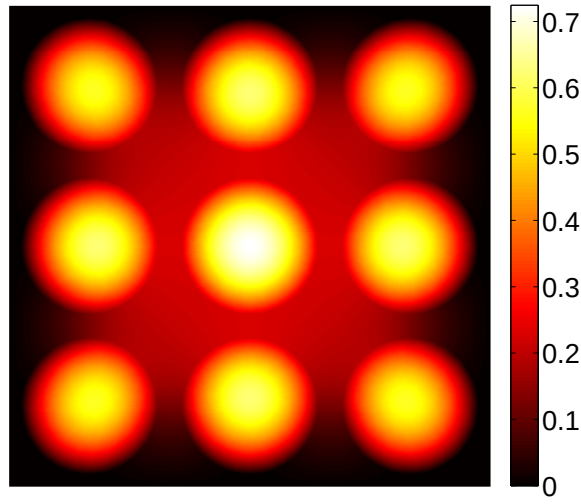
- **no reuse:** For each system, a new AMG hierachy is constructed.
- **center-based:** Reuse of the preconditioner constructed for the system at $\boldsymbol{\xi} = (0, \ldots, 0)$.
- **recycled setup:** AMG with recycled setup as described in §6.3.

We will run different experiments where each time one parameter is varied while the other parameters are set to their default value. These values are (brackets $[\cdot]$ denote the default)

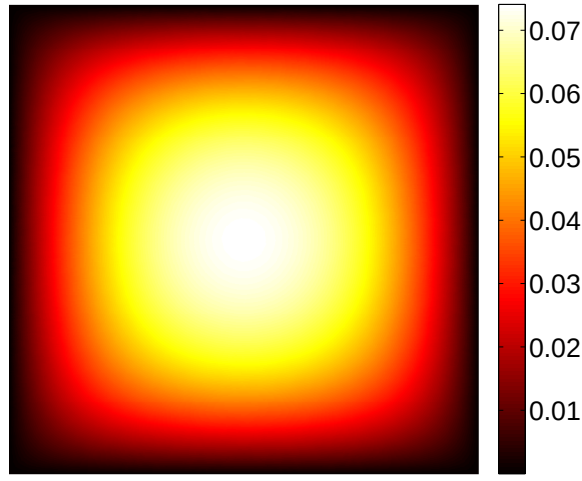
- **scaling factor** $\sigma = 0.02, 0.1, [0.49]$,
- **residual error** $\epsilon = [10^{-3}], 10^{-6}, 10^{-9}$ (i.e., CG is iterated until the relative residual error is below ϵ) ,
- **spatial nodes** $n_\phi = [4\,038], 12\,133, 34\,939$,
- **sparse grid level** $p = 1, [3], 5$,
- **fuzzy numbers** $d_\Xi = [4], 9, 16$.

As an illustration of the possibly large difference between the solutions of different systems, we plotted the solution for $\boldsymbol{\xi} = (-1, \ldots, -1)$ and $\boldsymbol{\xi} = (1, \ldots, 1)$ in figures 6.2a and 6.2b respectively ($n_\phi = 4\,038$, $d_\Xi = 9$, $\sigma = 0.49$).

Experiment 1, Table 6.1. Initial guess recycling strategies. The first experiment is an experiment where all parameters take their default value and where all combinations of initial guess and preconditioner recycling strategies are tested. In the other experiments, we will omit the second and third (i.e., "random" and "Gordon et al.") initial guess recycling strategy for the sake of brevity.



(a) $\boldsymbol{u}(-1,\ldots,-1)$



(b) $\boldsymbol{u}(1,\ldots,1)$

Figure 6.2: Solutions for $d_{\Xi} = 9$.

Table 6.1: Solution time in seconds and (in brackets) the mean number of CG iterations for the linear systems that arise during the construction of a sparse grid interpolating polynomial response surface approximation to the solution of the parameterized PDE corresponding to (6.1). The following parameter values are used: $n_\phi = 4\,038$; $d_\Xi = 4$; $\sigma = 0.49$; $p = 3$; $\epsilon = 10^{-3}$ (#systems = 137).

	Preconditioner					
Initial guess	**No reuse**		**Center-based**		**Recycled setup**	
Zeros	10.0	(8.3)	18.4	(17.6)	9.3	(8.4)
Random	11.4	(9.5)	27.2	(26.4)	10.4	(9.4)
Gordon et al.	9.4	(8.0)	21.0	(20.3)	8.9	(7.9)
Nearest	9.2	(7.3)	21.3	(20.6)	8.2	(7.2)

Table 6.1 lists the timings of this first experiment. We can observe that, for the "no reuse" and "recycled setup" preconditioner, a modest gain in performance is achieved when the "Gordon et al." or the "nearest" initial guess recycling strategies are applied (the "nearest" recycling strategy is slightly better). A lower number of CG iterations (and correspondingly a lower computation time) is needed to converge. This does not hold in the case of the center-based preconditioner. There, an all-zeros initial guess turns out to be the best choice. From the timings of the "random" recycling strategy, we can conclude that if an initial guess recycling strategy is used then the ordering of the systems is important. A bad ordering leads to a drop in performance in comparison to the standard all-zeros initial guess.

Another observation that we can make is that, despite a large scaling factor $\sigma = 0.49$, the preconditioner with recycled setup is as good as the "no reuse" preconditioner in terms of the number of iterations. The timings for the preconditioner with recycled setup are in fact slightly better. The reason for this is, however, not entirely clear to us. Concerning the center-based preconditioner, we can conclude that it performs poorly compared to the other preconditioners. In the experiments below, where the code is run for different parameter settings, we will see, however, that this is not always the case.

Experiment 2, Table 6.2a. Scaling factor $\sigma = 0.02, 0.1, 0.49$. For a small value $\sigma = 0.02$, we see that the center-based preconditioner performs as well as the other preconditioners in terms of the number of iterations. Thanks to a lower setup cost, the center-based preconditioner is therefore the fastest choice. Also for $\sigma = 0.1$, it is still the fastest despite a slightly higher number of iterations. Another observation is that the reward from recycling the initial guess gets higher the smaller the value of σ is. Remarkable maybe is that, for

a low value of σ, the center-based preconditioner benefits from recycling the initial guess, while the opposite is true for a high value of σ.

Experiment 3, Table 6.2b. Residual error $\epsilon = 10^{-3}, 10^{-6}, 10^{-9}$. For a large value $\epsilon = 10^{-3}$, the setup cost of the "no reuse" preconditioner dominates the total computation time. For smaller values of ϵ, this balance shifts, and we can see that the gain ratio from using the preconditoner with recycled setup decreases. Also the added benefit of recycling the initial guess becomes smaller.

Experiment 4, Table 6.2c. Spatial nodes $n_\phi = 4\,038, 12\,133, 34\,939$. The fact that the number of iterations for the "no reuse" preconditioner still increases for increasing values of n_ϕ indicates that the asymptotic $\mathcal{O}(n_\phi)$ behavior of multigrid has not been reached yet. We also see a slower than $\mathcal{O}(n_\phi)$ growth of the setup time of the preconditioner with recycled setup. The reason for this is not entirely clear, but it causes an increasing benefit of using the preconditioner with recycled setup.

Experiment 5, Table 6.2d. Sparse grid level $p = 1, 3, 5$. The number of systems to be solved changes with p. In fact, the average distance of the systems to the system in the origin increases. As a consequence, we can see an increasing number of iterations for the center-based preconditioner. Because these systems get closer to each other on average, we can observe an increasing benefit from recycling the initial guess for higher values of p. For $p = 5$, also the center-based preconditioner profits from the recycling of the initial guess.

Experiment 6, Table 6.2e. Fuzzy numbers $d_\Xi = 4, 9, 16$. The growth with the number of systems is faster than $\mathcal{O}(n_\psi)$. The problems for different values of d_Ξ are in fact hard to compare, because the solutions are quite different, i.e., $\boldsymbol{u}(\xi_1, \xi_2, \xi_3, \xi_4, 0, 0, 0, 0, 0) \neq \boldsymbol{u}(\xi_1, \xi_2, \xi_3, \xi_4)$. What we can observe, however, is that the advantage of the preconditioner with recycled setup over the center-based preconditioner decreases with a growing value of d_Ξ. The number of iterations for the center-based preconditioner almost drops as low as for the other preconditioners.

Table 6.2: Construction time of the preconditioner, solution time, and total computation time in seconds (in the format $\cdot + \cdot = \cdot$) and (in brackets) the mean number of CG iterations for the linear systems that arise during the construction of a sparse grid interpolating polynomial response surface approximation to the solution of the parametric PDE corresponding to (6.1).

(a) Parameters: $n_\phi = 4\,038$; $d_\Xi = 4$; $\sigma = 0.02, 0.1, 0.49$; $p = 3$; $\epsilon = 10^{-3}$ (#systems = 137).

	$\sigma = 0.02$	$\sigma = 0.1$	$\sigma = 0.49$
Preconditioner	**Initial guess: Zeros**		
No reuse	60.7 + 8.8 = 69.4 (8.0)	60.9 + 8.8 = 69.6 (8.0)	61.6 + 10.0 = 71.6 (8.3)
Center-based	0.4 + 9.0 = 9.4 (8.0)	0.4 + 9.3 = 9.8 (8.5)	0.6 + 18.3 = 18.9 (17.6)
Recycled setup	3.1 + 9.0 = 12.1 (8.0)	3.1 + 8.9 = 11.9 (8.0)	3.0 + 9.2 = 12.2 (8.4)
	Initial guess: Nearest		
No reuse	60.3 + 5.2 = 65.6 (4.3)	60.6 + 5.9 = 66.5 (5.1)	61.7 + 8.9 = 70.6 (7.3)
Center-based	0.5 + 5.5 = 5.9 (4.4)	0.4 + 6.2 = 6.6 (5.2)	0.5 + 21.2 = 21.7 (20.6)
Recycled setup	3.1 + 5.4 = 8.4 (4.4)	3.0 + 6.0 = 9.0 (5.1)	3.0 + 8.1 = 11.0 (7.2)

(b) Parameters: $n_\phi = 4\,038$; $d_\Xi = 4$; $\sigma = 0.49$; $p = 3$; $\epsilon = 10^{-3}, 10^{-6}, 10^{-9}$ (#systems = 137).

	$\epsilon = 10^{-3}$	$\epsilon = 10^{-6}$	$\epsilon = 10^{-9}$
Preconditioner	**Initial guess: Zeros**		
No reuse	61.5 + 9.9 = 71.4 (8.3)	61.7 + 16.2 = 77.9 (14.0)	62.1 + 22.4 = 84.5 (19.4)
Center-based	0.4 + 18.4 = 18.8 (17.6)	0.4 + 29.3 = 29.7 (28.6)	0.4 + 39.9 = 40.4 (39.5)
Recycled setup	3.1 + 9.4 = 12.5 (8.4)	3.2 + 15.1 = 18.3 (14.3)	3.1 + 20.5 = 23.6 (19.8)
	Initial guess: Nearest		
No reuse	62.6 + 8.9 = 71.5 (7.3)	62.3 + 15.3 = 77.6 (13.1)	62.0 + 21.1 = 83.1 (18.4)
Center-based	0.4 + 21.5 = 21.9 (20.6)	0.4 + 39.9 = 40.3 (39.2)	0.4 + 56.3 = 56.8 (56.0)
Recycled setup	3.1 + 8.2 = 11.3 (7.2)	3.1 + 13.9 = 17.0 (13.0)	3.1 + 19.2 = 22.3 (18.5)

(c) Parameters: $n_\phi = 4\,038, 12\,133, 34\,939$; $d_\Xi = 4$; $\sigma = 0.49$; $p = 3$; $\epsilon = 10^{-3}$ (#systems = 137).

	$n_\phi = 4\,038$	$n_\phi = 12\,133$	$n_\phi = 34\,939$
Preconditioner	**Initial guess: Zeros**		
No reuse	62.8 + 10.1 = 72.9 (8.3)	208.3 + 16.6 = 224.9 (9.8)	642.9 + 32.2 = 675.1 (11.2)
Center-based	0.4 + 18.2 = 18.7 (17.6)	1.5 + 34.0 = 35.5 (17.6)	4.6 + 71.7 = 76.2 (26.0)
Recycled setup	3.1 + 9.3 = 12.4 (8.4)	5.5 + 16.5 = 22.0 (10.2)	13.3 + 34.0 = 47.2 (11.7)
	Initial guess: Nearest		
No reuse	61.3 + 8.7 = 70.0 (7.3)	203.2 + 14.3 = 217.5 (8.5)	632.5 + 29.0 = 661.5 (10.2)
Center-based	0.4 + 21.3 = 21.7 (20.6)	1.5 + 39.5 = 41.0 (25.2)	4.6 + 89.4 = 94.0 (31.9)
Recycled setup	3.1 + 8.2 = 11.3 (7.2)	5.6 + 14.0 = 19.6 (8.5)	12.9 + 29.7 = 42.7 (10.4)

(d) Parameters: $n_\phi = 4\,038$; $d_\Xi = 4$; $\sigma = 0.49$; $p = 1, 3, 5$; $\epsilon = 10^{-3}$.

	$p = 1$ (#systems = 9)	$p = 3$ (#systems = 137)	$p = 5$ (#systems = 1105)
Preconditioner	**Initial guess: Zeros**		
No reuse	4.1 + 0.6 = 4.7 (8.2)	61.2 + 10.0 = 71.2 (8.3)	496.7 + 83.2 = 579.9 (8.3)
Center-based	0.4 + 1.0 = 1.5 (14.2)	0.4 + 18.3 = 18.8 (17.6)	0.5 + 169.0 = 169.5 (18.2)
Recycled setup	0.6 + 0.6 = 1.2 (8.2)	3.1 + 9.6 = 12.6 (8.4)	24.8 + 81.3 = 106.1 (8.5)
	Initial guess: Nearest		
No reuse	4.1 + 0.6 = 4.7 (7.8)	61.0 + 8.9 = 69.8 (7.3)	497.5 + 71.6 = 569.1 (6.9)
Center-based	0.4 + 1.3 = 1.7 (18.1)	0.4 + 22.0 = 22.5 (20.6)	0.4 + 140.8 = 141.3 (15.0)
Recycled setup	0.6 + 0.6 = 1.2 (7.8)	3.1 + 8.2 = 11.3 (7.2)	25.3 + 68.4 = 93.7 (6.8)

(e) Parameters: $n_\phi = 4\,038$; $d_\Xi = 4, 9, 16$; $\sigma = 0.49$; $p = 3$; $\epsilon = 10^{-3}$.

	$d_\Xi = 4$ (#systems = 137)	$d_\Xi = 9$ (#systems = 1177)	$d_\Xi = 16$ (#systems = 6049)
Preconditioner	**Initial guess: Zeros**		
No reuse	62.2 + 10.6 = 72.8 (8.3)	726.4 + 135.8 = 862.2 (9.0)	3023.3 + 984.2 = 4007.5 (8.1)
Center-based	0.4 + 19.4 = 19.8 (17.6)	0.6 + 225.8 = 226.4 (16.1)	0.5 + 1324.6 = 1325.1 (11.7)
Recycled setup	3.2 + 9.8 = 13.0 (8.4)	41.1 + 138.3 = 179.4 (9.2)	390.9 + 971.4 = 1362.3 (8.1)
	Initial guess: Nearest		
No reuse	62.4 + 8.8 = 71.2 (7.3)	728.1 + 117.7 = 845.8 (7.6)	3076.4 + 931.3 = 4006.7 (6.7)
Center-based	0.4 + 22.3 = 22.8 (20.6)	0.6 + 340.6 = 341.2 (24.8)	0.5 + 2458.3 = 2458.8 (23.8)
Recycled setup	3.2 + 8.7 = 11.9 (7.2)	39.3 + 119.7 = 159.0 (7.8)	374.8 + 813.3 = 1188.1 (6.8)

6.5 Conclusions

This chapter presented two recycling strategies to speed up the solution of a long sequence of similar linear systems. Such sequences of systems can arise in the construction of a response surface approximation by a collocation approach or the direct computation of quantities of interest by the α-cut approach of a fuzzy partial differential equation. The idea of recycling is that intermediate results or computations of previously solved systems can be used to reduce the computational effort of solving the next system. Most of the recycling strategies found in the literature are based on the premise that an iterative solver is used.

In the numerical experiments that we presented, we see that the initial guess recycling strategy yields a very consistent, though very modest, reduction of the number of needed Krylov iterations in the case of the full AMG preconditioner and the AMG preconditioner with recycled setup. When a center-based preconditioner is used, the results are far less consistent, and often show a large increase of the number of iterations.

The full AMG preconditioner was found to be always much more expensive than the center-based preconditioner (regularly about 3 to 4 times more expensive in total computation time for small spatial grids, and up to about a factor 10 for larger grids). Only when a very high accuracy of the solution would be required, it could pay off to use such an expensive (though excellent) preconditioner. The AMG preconditioner with recycled setup on the other hand, was able to outperform the center-based preconditioner in most of the experiments, especially when combined with the initial guess recycling strategy (with speedup factors of about 2). This is due to its much lower setup cost than needed for the full AMG preconditioner and its better quality than the center-based preconditioner. The number of iterations needed in case of the AMG preconditioner with recycled setup was in fact always nearly equal to the number of iterations needed in case of the full AMG preconditioner. The only cases where the center-based preconditioner was the best choice, were observed when the value of the scaling factor σ was small. In that case, all the systems are very similar to the system in the center.

Chapter 7

Conclusions

This chapter presents the main results of this thesis and gives an overview of possible future research directions. The main contributions can be summarized as follows:

- an extensive comparison of the interpretation and definition of fuzzy DEs based on the Hukuhara derivative and some of its variants, differential inclusions, and sample path-based fuzzy fields; a clear motivation in the light of the possibilistic interpretation of fuzzy sets and a consistent mathematical framework for the sample path-based approach;
- proof that a response surface solution to the parametric DE which is accurate in the L^∞-norm over the parameter domain allows for an accurate (in the supremum distance d_∞) computation of certain fuzzy quantities of interest of the fuzzy DE;
- the modeling of a fuzzy field in similarity to the Karhunen–Loève expansion of a stochastic field;
- an overview of numerical procedures and corresponding error estimates in the L^∞-norm that can be found in the literature for the solution of the parametric PDE corresponding to the fuzzy diffusion equation with a focus on the collocation and the Galerkin approach;
- construction of a center-based and a multigrid preconditioner for the linear system that results from a Galerkin discretization of the parametric diffusion equation using Chebyshev polynomials in the parameter domain;

- local Fourier analysis of the multigrid preconditioner followed by an extensive numerical study of the diffusion equation and an elasticity problem;
- a fuzzification procedure consisting of two stages: the construction based on a low-rank tensor approximation of the function on a grid followed by the search for the minimal and maximal entries in the low-rank tensor;
- two recycling techniques for the solution of sequences of similar linear systems: reuse of a previous solution as initial guess for the next system to solve with a particular ordering of the systems, and recycling of the intergrid transfer operators of the multigrid preconditioner of a reference system.

7.1 Summary and conclusions

Uncertainty quantification is playing an increasingly important role in the mathematical modeling of physical phenomena. In engineering applications, for example, an accurate assessment of the uncertainties is necessary to make an informed guess of the reliability of a design, to do a risk analysis, to make a robust design, etcetera. Uncertainties come in different forms and from different sources. The classical way to model all these kinds of uncertainties is to use probability theory. Information or knowledge about a phenomenon may, however, be incomplete, contradicting, vague, etcetera. In that case, it can be argued that probability may not be the most appropriate mathematical representation of the uncertainty. In this thesis, we confined ourselves to uncertainties modeled by fuzzy sets. Fuzzy sets were first proposed as a way to model vague linguistic knowledge in logic. It was, however, soon realized that the membership function of a fuzzy set can also be regarded as a possibility distribution. As such, a fuzzy set can be interpreted as an imprecise probability.

Fuzzy differential equations

The objective of this thesis was to develop and analyze numerical methods for solving fuzzy differential equations. The main model problem that we considered for this was the steady-state diffusion equation with a fuzzy diffusion coefficient. A few different interpretations and definitions of fuzzy DEs can be found in the literature. We gave an overview of the most important ones: the one based on the Hukuhara derivative and some of its variants, the one based on differential inclusions, and finally the one based on sample path-based fuzzy fields (which includes the approach followed in the so-called fuzzy

finite element method). The latter one seems the most natural and intuitive way of defining fuzzy DEs when adhering to the possibilistic interpretation of fuzzy sets. Defining fuzzy processes and fields as fuzzy sets over the space of distributions or some Sobolev space allows for a very natural study of fuzzy differential equations. The definition of the solution to such a fuzzy DE is then simply defined as the fuzzification by Zadeh's extension principle of the solution operator of a corresponding parametric differential equation.

Computing quantities of interest defined by a fuzzy DE by the α-cut approach amounts to solving a sequence of DE-constrained global optimization problems over nested search spaces. The large amount of information shared by these optimization problems leaves room for many algorithmic improvements over the independent treatment of these problems. A first strategy we discussed was the response surface approach, which means that, prior to the computation of the quantities of interest, a response surface approximation of the solution to the parametric differential equation is constructed. We showed that if such a response surface solution is accurate in the L^∞-norm over the parameter domain, then it can be used to accurately (in the supremum distance d_∞ for fuzzy sets) compute certain fuzzy quantities of interest.

We continued with an overview from the literature of some numerical procedures and corresponding error estimates in the L^∞-norm for the solution of the diffusion equation where the fuzzy diffusion coefficient is modeled in a way similar to the Karhunen–Loève expansion of a stochastic field. More particularly, we focused on the collocation and the Galerkin approach which employ a spectral polynomial discretization in the parameter domain. The error estimates show that an algebraic convergence as a function of the number of polynomial basis functions can be achieved even in the infinite-dimensional case where the parameter domain $\Xi := [\widetilde{\boldsymbol{\xi}}]_0 = [-1, 1]^\infty$ if the domain of analyticity of the solution in the parameter domain grows sufficiently with the dimension.

Preconditioners for the Galerkin system

The Galerkin discretization of the parametric diffusion equation results in a large linear algebraic system of Kronecker product structure. The computational aspects of solving this system in the case where Chebyshev polynomials are used in the parameter domain was the topic of Chapter 4. Based on ideas from the literature on stochastic PDEs, we proposed a center-based preconditioner and a multigrid preconditioner. By means of a local Fourier analysis, we showed that the convergence properties of the preconditioners are optimal w.r.t. the discretization parameters, i.e., the size of the spatial mesh and the number of Chebyshev polynomials. The performed

numerical experiments demonstrated an exponential convergence of the Galerkin approximation as a function of the polynomial degree. Furthermore, the experiments confirmed the theoretical results of the local Fourier analysis and demonstrated a high robustness of the multigrid preconditioner to the magnitude of the input uncertainty. An almost constant number of iterations was observed over a very wide range of parameter values. The center-based preconditioner on the other hand turned out to be the most efficient solver overall.

Low-rank tensor based methods for fuzzification

In Chapter 5, we made abstraction of the underlying problem of solving a fuzzy DE and treated the general problem of computing a function of fuzzy numbers by the α-cut approach. If the function is real-valued and continuous, and if the fuzzy numbers are noninteractive, then the problem amounts to solving a sequence of global minimization and global maximization problems over nested hyperrectangles. Without extra assumptions on the structure of the function except for continuity, the global optimization problems are known to scale exponentially in complexity with the dimension. We made the assumption that the function has some sort of low-rank structure. For such functions, we proposed a derivative-free fuzzification algorithm based on a low-rank tensor approximation of the function on a grid followed by the search for the minimal and maximal entries in the low-rank tensor. As such, exchange of information between the different optimization problems in the α-cut approach is established by the shared construction of the low-rank tensor approximation.

We ended this chapter with a comparison of the proposed fuzzification algorithm and some state-of-the-art global optimization routines in a numerical test containing some challenging high-dimensional optimization problems. This showed us that there is a large potential in the use of low-rank tensor methods in optimization and, more generally, fuzzification. Whereas state-of-the-art global optimization algorithms such as MCS and GLCCLUSTER have an increasing difficulty to deal with higher dimensional search spaces, the low-rank approximation approach seems to effectively break the curse of dimensionality for the functions we tested. Despite the promising results, theoretical results of convergence, complexity and stability of both the low-rank tensor approximation algorithm and the algorithm which finds the extremal elements in a low-rank tensor are, however, still largely missing.

Recycling strategies for the solution of sequences of similar linear systems

Chapter 6 presented two recycling strategies to speed up the solution of a long sequence of similar linear systems. Such sequences of systems arise, for example, in the collocation approach to construct a response surface approximation of the solution to the parametric diffusion equation. The idea of recycling is that intermediate results or computations of previously solved systems can be used to reduce the computational effort of solving the next system. A recycling method we proposed is the one in which the prolongation and restriction operators of the multigrid preconditioner of one reference system are recycled to the other systems. In an algebraic multigrid (AMG) procedure, the construction of these intergrid transfer operators can amount to a significant portion of the total computation time. By recycling them, a preconditioner can be constructed which is nearly as good as a finely tuned AMG preconditioner while being much cheaper to construct. A second recycling strategy that we discussed is the one in which a previous solution is reused as an initial guess for solving the next system. In that regard, it is important to solve the systems in a certain order such that earlier solutions can provide a good initial guess to subsequent ones. We proposed such an ordering strategy.

In the numerical experiments that we presented, we see that the initial guess recycling strategy yields a very consistent, though very modest, reduction of the number of needed Krylov iterations in the case of the full AMG preconditioner and the AMG preconditioner with recycled setup. When a center-based preconditioner is used, the results are far less consistent, and often show a large increase of the number of iterations. The full AMG preconditioner was found to be always much more expensive than the center-based preconditioner. Only when a very high accuracy of the solution would be required, it could pay off to use such an expensive (though excellent) preconditioner. The AMG preconditioner with recycled setup on the other hand, was able to outperform the center-based preconditioner in most of the experiments, especially when combined with the initial guess recycling strategy.

7.2 Suggestions for future research

Tensor grid collocation and tensor product Galerkin

The tensor grid collocation and tensor product Galerkin discretization procedures lead to a tensor-structured algebraic system. The cost of solving such systems using traditional matrix-based methods rapidly becomes too high for higher dimensions. Much research effort is currently going to solving such

systems using low-rank tensor based methods [8, 60, 70, 114, 115, 113, 126, 125, 145, 167]. Although very promising, there seems to be a lot of room for improvement on various levels. Moreover, the current methods are often applied in an ad-hoc manner. Theoretical results are still rather scarce.

Polynomial optimization

As mentioned in Chapter 5, the information-based complexity of derivative-free optimization is lower than the complexity of the approximation problem. The problem when a polynomial approximation is constructed is, however, that the optimization of this polynomial could still be very costly. There are many approaches to optimizing a polynomial and one of them is the sums-of-squares approach [130, 131, 169, 170, 171]. There are only few results on the complexity of the sums-of-squares optimization of polynomials with specific kinds of structures. It would, for example, be interesting to gain more insight in the complexity of optimizing polynomials with a low-rank tensor structure. Further, in the context of fuzzifying a function by the α-cut approach, it should be investigated how the similarity of the polynomial optimization on the different nested search spaces could be exploited.

L^∞ error estimates for the Galerkin approach

The natural norm in which the error of the approximation by a Galerkin discretization (in the parameter domain) is estimated is of L^2-type. The L^2_w projection (with Chebyshev weight or Legendre weight, for example) of a continuous function on a polynomial approximation space results in a quasi-optimal approximation in the L^∞-norm. It seems therefore that similar positive results would hold for the Galerkin projection if the diffusion coefficient satisfies the right smoothness conditions.

Fuzzy fields

Our model for the fuzzy diffusion coefficient was artificially constructed from a Karhunen–Loève expansion of a stochastic field. How such a model for fuzzy fields or processes would be constructed in a real-life situation from measurements, expert knowledge, etcetera, is, however, still a topic of ongoing research [148, 153, 203]. It seems that in that regard a better understanding of the connection between the sample path view on fuzzy fields and its counterpart, the fuzzy-valued view, could be of importance.

Combination of different types of uncertainty

The only type of uncertainty treated in this thesis was fuzzy uncertainty. Although combining different types of uncertainty models like probability, Dempster–Shafer uncertainty, fuzzy sets, is not something new [12, 67, 94, 140], there seems to be a lot of room for further improvements and research.

Black-box low-rank tensor approximation; finding extremal elements in low-rank tensor

Despite the promising results of the numerical experiments in Chapter 5, theoretical results of convergence, complexity and stability of both the low-rank tensor approximation algorithm and the algorithm which finds the extremal elements in a low-rank tensor are still missing.

AMG with recycled preconditioner

The AMG with recycled preconditioner performed really well in the numerical experiments of Chapter 6. Also here, a better theoretical understanding of its performance, when it would be beneficial to apply, when it is better to construct new intergrid transfer operators than recycling previous ones, would be welcome. Another interesting research direction would be to see how a Krylov subspace recycling technique [28, 116, 168, 189, 211] would interact with the AMG with recycled preconditioner.

Bibliography

[1] Adhikari, S., and Khodaparast, H. H. A spectral approach for fuzzy uncertainty propagation in finite element analysis. *Fuzzy Sets and Systems* (2013).

[2] Adler, R. J., and Hasofer, A. M. *The geometry of random fields*. Wiley, New York, 1981.

[3] Akpan, U., Koko, T., Orisamolu, I., and Gallant, B. Practical fuzzy finite element analysis of structures. *Finite Elements in Analysis and Design 38* (2001), 23–111.

[4] Babuška, I., Nobile, F., and Tempone, R. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Journal on Numerical Analysis 45*, 3 (2007), 1005–1034.

[5] Babuška, I., Tempone, R., and Zouraris, G. E. Galerkin finite element approximations of stochastic elliptic partial differential equations. *SIAM Journal on Numerical Analysis 42*, 2 (2004), 800–825.

[6] Babuška, I., Tempone, R., and Zouraris, G. E. Solving elliptic boundary value problems with uncertain coefficients by the finite element method: the stochastic formulation. *Computer Methods in Applied Mechanics and Engineering 194*, 12-16 (2005), 1251–1294.

[7] Bäck, J., Nobile, F., Tamellini, L., and Tempone, R. Stochastic spectral Galerkin and collocation methods for PDEs with random coefficients: a numerical comparison. In *Spectral and High Order Methods for Partial Differential Equations*, J. S. Hesthaven and E. M. Rønquist, Eds., vol. 76 of *Lecture Notes in Computational Science and Engineering*. Springer Berlin Heidelberg, Berlin, 2011, pp. 43–62.

[8] Ballani, J., and Grasedyck, L. A projection method to solve linear systems in tensor format. *Numerical Linear Algebra With Applications 20* (2013), 27–43.

[9] Ballani, J., Grasedyck, L., and Kluge, M. Black box approximation of tensors in hierarchical Tucker format. *Linear Algebra and its Applications 438*, 2 (2013), 639–657.

[10] Barros, L. C., Gomes, L. T., and Tonelli, P. A. Fuzzy differential equations: an approach via fuzzification of the derivative operator. *Fuzzy Sets and Systems 230* (2013), 39–52.

[11] Barthelmann, V., Novak, E., and Ritter, K. High dimensional polynomial interpolation on sparse grids. *Advances in Computational Mathematics 12* (2000), 273–288.

[12] Baudrit, C., Couso, I., and Dubois, D. Joint propagation of probability and possibility in risk analysis: towards a formal framework. *International Journal of Approximate Reasoning 45*, 1 (2007), 82–105.

[13] Bebendorf, M. Approximation of boundary element matrices. *Numerische Mathematik 86* (2000), 565–589.

[14] Bede, B., and Gal, S. G. Generalizations of the differentiability of fuzzy-number-valued functions with applications to fuzzy differential equations. *Fuzzy Sets and Systems 151*, 3 (2005), 581–599.

[15] Bede, B., and Stefanini, L. Generalized differentiability of fuzzy-valued functions. *Fuzzy Sets and Systems 230* (2013), 119–141.

[16] Bellman, R. *Dynamic programming*. Rand Corporation research study. Princeton University Press, Princeton, NJ, 1957.

[17] Bieri, M., and Schwab, C. Sparse high order FEM for elliptic sPDEs. *Computer Methods in Applied Mechanics and Engineering 198*, 13-14 (2009), 1149–1170.

[18] Binev, P., Cohen, A., Dahmen, W., Devore, R., Petrova, G., and Wojtaszczyk, P. Convergence rates for greedy algorithms in reduced basis methods. *SIAM Journal on Mathematical Analysis 43*, 3 (2011), 1457–1472.

[19] Börm, S., and Grasedyck, L. Hybrid cross approximation of integral operators. *Numerische Mathematik 101*, 2 (2005), 221–249.

[20] Boyd, J. P. *Chebyshev and Fourier spectral methods*. Courier Dover Publications, New York, 2001.

[21] Butler, T., Constantine, P., and Wildey, T. A posteriori error analysis of parameterized linear systems using spectral methods. *SIAM Journal on Matrix Analysis and Applications 33*, 1 (2012), 195–209.

[22] Canuto, C., Hussaini, M. Y., Quarteroni, A., and Zang, T. A. *Spectral methods: fundamentals in single domains.* Springer-Verlag, Berlin, 2006.

[23] Carroll, J. D., and Chang, J. J. Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart–Young" decomposition. *Psychometrika 35*, 3 (1970), 283–319.

[24] Chalco-Cano, Y., Misukoshi, M. T., Román-Flores, H., and Flores-Franulic, A. Spline approximation for Zadeh's extensions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 17*, 02 (2009), 269–280.

[25] Chalco-Cano, Y., and Román-Flores, H. On new solutions of fuzzy differential equations. *Chaos, Solitons & Fractals 38*, 1 (2008), 112–119.

[26] Chalco-Cano, Y., and Román-Flores, H. Comparation between some approaches to solve fuzzy differential equations. *Fuzzy Sets and Systems 160*, 11 (2009), 1517–1527.

[27] Chalco-Cano, Y., and Román-Flores, H. Some remarks on fuzzy differential equations via differential inclusions. *Fuzzy Sets and Systems 230* (2013), 3–20.

[28] Chan, T. F., and Ng, M. K. Galerkin projection methods for solving multiple linear systems. *SIAM Journal on Scientific Computing 21*, 3 (1999), 836–850.

[29] Chen, L., and Rao, S. S. Fuzzy finite-element approach for the vibration analysis of imprecisely-defined systems. *Finite Elements in Analysis and Design 27*, 1 (1997), 69–83.

[30] Chkifa, A., Cohen, A., DeVore, R., and Schwab, C. Sparse adaptive Taylor approximation algorithms for parametric and stochastic elliptic PDEs. *Modélisation Mathématique et Analyse Numérique 47*, 1 (2013), 253–280.

[31] Chkifa, A., Cohen, A., and Schwab, C. High-dimensional adaptive sparse polynomial interpolation and applications to parametric PDEs. *Foundations of Computational Mathematics* (2013), 1–33.

[32] Cohen, A., DeVore, R., Petrova, G., and Wojtaszczyk, P. Finding the minimum of a function. Tech. rep., 2013.

[33] Cohen, A., DeVore, R., and Schwab, C. Convergence rates of best N-term Galerkin approximations for a class of elliptic sPDEs. *Foundations of Computational Mathematics 10*, 6 (2010), 615–646.

[34] Cohen, A., Devore, R., and Schwab, C. Analytic regularity and polynomial approximation of parametric and stochastic elliptic PDE's. *Analysis and Applications 09*, 01 (2011), 11–47.

[35] Constantine, P. G., Gleich, D. F., and Iaccarino, G. Spectral methods for parameterized matrix equations. *SIAM Journal on Matrix Analysis and Applications 31*, 5 (2010), 2681–2699.

[36] Corveleyn, S., Rosseel, E., and Vandewalle, S. Iterative solvers for a spectral Galerkin approach to elliptic partial differential equations with fuzzy coefficients. *SIAM Journal on Scientific Computing 35*, 5 (2013), S420–S444.

[37] Corveleyn, S., and Vandewalle, S. Component reuse in iterative solvers for the solution of fuzzy partial differential equations. In *Proceedings of the 7th International Conference of Numerical Analysis and Applied Mathematics* (Rethymno, Greece, 2009), pp. 452–455.

[38] Corveleyn, S., and Vandewalle, S. On the numerical solution of fuzzy elliptic PDEs by means of polynomial response surfaces. In *Proceedings of the International Conference on Noise and Vibration Engineering* (Leuven, Belgium, 2010), pp. 5007–5014.

[39] Corveleyn, S., and Vandewalle, S. A polynomial reponse surface approach for the solution of fuzzy elliptic partial differential equations. In *Proceedings of the 8th International Conference on Structural Dynamics* (Leuven, Belgium, 2011), pp. 3048–3055.

[40] Cox, R. T. *The algebra of probable inference*. John Hopkins Press, Baltimore, MD, 1961.

[41] de Cooman, G. Possibility theory I: the measure- and integral-theoretic groundwork. *International Journal of General Systems 25*, 4 (1997), 291–323.

[42] de Cooman, G. Possibility theory II: conditional possibility. *International Journal of General Systems 25*, 4 (1997), 325–351.

[43] de Cooman, G. Possibility theory III: possibilistic independence. *International Journal of General Systems 25*, 4 (1997), 353–371.

[44] de Cooman, G., and Aeyels, D. On the coherence of supremum preserving upper previsions. In *Proceedings of the 6th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems* (Granada, Spain, 1996), B. Bouchon-Meunier, Ed., pp. 1405–1410.

[45] de Cooman, G., and Aeyels, D. Supremum preserving upper probabilities. *Information Sciences 118* (1999), 173–212.

[46] de Cooman, G., and Kerre, E. E. Ample fields. *Simon Stevin: A Quarterly Journal of Pure and Applied Mathematics 67* (1993), 235–244.

[47] De Finetti, B. Foresight: its logical laws, its subjective sources. In *Subjective probability*, H. E. Kyburg and H. E. Smokler, Eds. Wiley, New York, 1964, pp. 93–158.

[48] De Finetti, B. *Theory of probability*, vol. 1. John Wiley & Sons Ltd., New York, 1974.

[49] de Klerk, E. The complexity of optimizing over a simplex, hypercube or sphere: a short survey. *Central European Journal of Operations Research 16*, 2 (2008), 111–125.

[50] de Klerk, E., and Laurent, M. Error bounds for some semidefinite programming approaches to polynomial minimization on the hypercube. *SIAM Journal on Optimization 20*, 6 (2010), 3104–3120.

[51] De Munck, M., Moens, D., Desmet, W., and Vandepitte, D. A response surface based optimisation algorithm for the calculation of fuzzy envelope FRFs of models with uncertain properties. *Computers & Structures 86*, 10 (2008), 1080–1092.

[52] Degrauwe, D. *Uncertainty propagation in structural analysis by fuzzy numbers*. PhD thesis, University of Leuven, Leuven, Belgium, 2007.

[53] Dempster, A. P. Upper and lower probabilities induced by a multivalued mapping. *The Annals of Mathematical Statistics 38*, 2 (1967), 325–339.

[54] Destercke, S., Dubois, D., and Chojnacki, E. Unifying practical uncertainty representations: I. Generalized p-boxes. *International Journal of Approximate Reasoning 49*, 3 (2008), 649–663.

[55] Diamond, P. Time-dependent differential inclusions, cocycle attractors and fuzzy differential equations. *IEEE Transactions on Fuzzy Systems 7*, 6 (1999), 734–740.

[56] Diamond, P. Stability and periodicity in fuzzy differential equations. *IEEE Transactions on Fuzzy Systems 8*, 5 (2000), 583–590.

[57] Diamond, P., and Kloeden, P. Metric spaces of fuzzy sets. *Fuzzy Sets and Systems 100* (1999), 63–71.

[58] Donders, S., Vandepitte, D., Van de Peer, J., and Desmet, W. Assessment of uncertainty on structural dynamic responses with the short transformation method. *Journal of Sound and Vibration 288*, 3 (2005), 523–549.

[59] Dong, W. M., and Wong, F. S. Fuzzy weighted averages and implementation of the extension principle. *Fuzzy Sets and Systems 21* (1987), 183–199.

[60] Doostan, A., and Iaccarino, G. A least-squares approximation of partial differential equations with high-dimensional random inputs. *Journal of Computational Physics 228*, 12 (2009), 4332–4345.

[61] Dubois, D. Possibility theory and statistical reasoning. *Computational Statistics & Data Analysis 51*, 1 (2006), 47–69.

[62] Dubois, D., and Prade, H. Towards fuzzy differential calculus part 1: integration of fuzzy mappings. *Fuzzy sets and Systems 8* (1982), 1–17.

[63] Dubois, D., and Prade, H. Towards fuzzy differential calculus part 2: integration on fuzzy intervals. *Fuzzy Sets and Systems 8* (1982), 105–115.

[64] Dubois, D., and Prade, H. Towards fuzzy differential calculus part 3: differentiation. *Fuzzy sets and systems 8* (1982), 225–233.

[65] Dubois, D., and Prade, H. The three semantics of fuzzy sets. *Fuzzy Sets and Systems 90*, 2 (1997), 141–150.

[66] Dubois, D., and Prade, H. Possibility theory, probability theory and multiple-valued logics: a clarification. *Annals of Mathematics and Artificial Intelligence 32*, 1-4 (2001), 35–66.

[67] Eldred, M. S., Swiler, L. P., and Tang, G. Mixed aleatory-epistemic uncertainty quantification with stochastic expansions and optimization-based interval estimation. *Reliability Engineering & System Safety 96*, 9 (2011), 1092–1113.

[68] Elman, H., and Furnival, D. Solving the stochastic steady-state diffusion problem using multigrid. *IMA Journal of Numerical Analysis 27*, 4 (2007), 675–688.

[69] Elman, H. C., Miller, C. W., Phipps, E. T., and Tuminaro, R. S. Assessment of collocation and Galerkin approaches to linear diffusion equations with random data. *International Journal for Uncertainty Quantification 1*, 1 (2011), 19–33.

[70] Espig, M., Hackbusch, W., Litvinenko, A., Matthies, H. G., and Wähnert, P. Efficient low-rank approximation of the stochastic Galerkin matrix in tensor formats. *Computers & Mathematics with Applications 67*, 4 (2012), 818–829.

[71] Espig, M., Hackbusch, W., Litvinenko, A., Matthies, H. G., and Zander, E. Efficient analysis of high dimensional data in tensor formats. In *Sparse grids and applications*, J. Garcke and M. Griebel, Eds., vol. 88 of *Lecture Notes in Computational Science and Engineering*. Springer, Berlin, 2013, pp. 31–56.

[72] Farkas, L. *An efficient fuzzy non-deterministic approach for structural finite element analysis*. PhD thesis, University of Leuven, Leuven, Belgium, 2012.

[73] Fischer, P. F. Projection techniques for iterative solution of Ax = b with successive right-hand sides. *Computer Methods in Applied Mechanics and Engineering 163* (1998), 193–204.

[74] Foo, J., Wan, X., and Karniadakis, G. E. The multi-element probabilistic collocation method (ME-PCM): error analysis and applications. *Journal of Computational Physics 227*, 22 (2008), 9572–9595.

[75] Frauenfelder, P., Schwab, C., and Todor, R. A. Finite elements for elliptic problems with stochastic coefficients. *Computer Methods in Applied Mechanics and Engineering 194*, 2-5 (2005), 205–228.

[76] Ganapathysubramanian, B., and Zabaras, N. Sparse grid collocation schemes for stochastic natural convection problems. *Journal of Computational Physics 225*, 1 (2007), 652–685.

[77] Ghanem, R. G., and Spanos, P. D. *Stochastic finite elements: a spectral approach*. Springer-Verlag, New York, 1991.

[78] Gittelson, C. J. Uniformly convergent adaptive methods for a class of parametric operator equations. *ESAIM: Mathematical Modelling and Numerical Analysis 46*, 6 (2012), 1485–1508.

[79] Gittelson, C. J. An adaptive stochastic Galerkin method for random elliptic operators. *Mathematics of Computation 82*, 283 (2013), 1515–1541.

[80] Goetschel, R., and Voxman, W. Elementary fuzzy calculus. *Fuzzy Sets and Systems 18* (1986), 31–43.

[81] Gordon, A., and Powell, C. E. On solving stochastic collocation systems with algebraic multigrid. *IMA Journal of Numerical Analysis 32*, 3 (2012), 1051–1070.

[82] Goreinov, S. A., Tyrtyshnikov, E. E., and Zamarashkin, N. L. A theory of pseudoskeleton approximations. *Linear Algebra and its Applications 261* (1997), 1–21.

[83] Grasedyck, L. Hierarchical singular value decomposition of tensors. *SIAM Journal on Matrix Analysis and Applications 31*, 4 (2010), 2029–2054.

[84] Grasedyck, L., Kressner, D., and Tobler, C. A literature survey of low-rank tensor approximation techniques. *arXiv preprint arXiv:1302.7121* (2013), 1–20.

[85] Gutmann, H.-M. A radial basis function method for global optimization. *Journal of Global Optimization 19* (2001), 201–227.

[86] Haasdonk, B., Urban, K., and Wieland, B. Reduced basis methods for parametrized partial differential equations with stochastic influences using the Karhunen-Loeve expansion. *SIAM/ASA Journal on Uncertainty Quantification 1* (2013), 79–105.

[87] Hackbusch, W. L^∞ estimation of tensor truncations. *Numerische Mathematik* (2012), 1–22.

[88] Hackbusch, W., and Kühn, S. A new scheme for the tensor representation. *Journal of Fourier Analysis and Applications 15*, 5 (2009), 706–722.

[89] Hansen, E. Global optimization using interval analysis—The multi-dimensional case. *Numerische Mathematik 34* (1980), 247–270.

[90] Hansen, E., and Walster, W. G. *Global optimization using interval analysis*. Marcel Dekker, Inc., New York, 2004.

[91] Hanss, M. The transformation method for the simulation and analysis of systems with uncertain parameters. *Fuzzy Sets and Systems 130*, 3 (2002), 277–289.

[92] Hanss, M. The extended transformation method for the simulation and analysis of fuzzy-parameterized models. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 11*, 6 (2003), 711–727.

[93] Harshman, R. A. Foundations of the PARAFAC procedure: models and conditions for an "explanatory" multimodal factor analysis. *UCLA Working Papers in Phonetics 16* (1970), 1–84.

[94] Helton, J., Johnson, J., Oberkampf, W. L., and Sallaberry, C. J. Representation of analysis results involving aleatory and epistemic uncertainty. *International Journal of General Systems 39*, 6 (2010), 605–646.

[95] Helton, J. C., and Johnson, J. D. Quantification of margins and uncertainties: alternative representations of epistemic uncertainty. *Reliability Engineering & System Safety 96*, 9 (2011), 1034–1052.

[96] Helton, J. C., Johnson, J. D., and Oberkampf, W. L. An exploration of alternative approaches to the representation of uncertainty in model predictions. *Reliability Engineering & System Safety 85*, 1–3 (2004), 39–71.

[97] Helton, J. C., Johnson, J. D., Sallaberry, C. J., and Storlie, C. B. Survey of sampling-based methods for uncertainty and sensitivity analysis. *Reliability Engineering & System Safety 91*, 10-11 (2006), 1175–1209.

[98] Horn, M. Optimal algorithms for global optimization in case of unknown Lipschitz constant. *Journal of Complexity 22*, 1 (2006), 50–70.

[99] Huang, D., Allen, T. T., Notz, W. I., and Zeng, N. Global optimization of stochastic black-box systems via sequential kriging meta-models. *Journal of Global Optimization 34*, 3 (2006), 441–466.

[100] Huckle, T., Waldherr, K., and Schulte-Herbrüggen, T. Computations in quantum tensor networks. *Linear Algebra and its Applications 438*, 2 (2013), 750–781.

[101] Hüllermeier, E. An approach to modelling and simulation of uncertain dynamical systems. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 2*, 2 (1997), 117–137.

[102] Huyer, W., and Neumaier, A. Global optimization by multilevel coordinate search. *Journal of Global Optimization 14* (1999), 331–355.

[103] Jakeman, J., Eldred, M., and Xiu, D. Numerical approach for quantification of epistemic uncertainty. *Journal of Computational Physics 229*, 12 (2010), 4648–4663.

[104] Jakeman, J. D., Narayan, A., and Xiu, D. Minimal multi-element stochastic collocation for uncertainty quantification of discontinuous functions. *Journal of Computational Physics 242* (2013), 790–808.

[105] Jeffreys, H. *The theory of probability.* Oxford University Press, Oxford, UK, 1998.

[106] Jones, D. R. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization 21* (2001), 345–383.

[107] Jones, D. R., Perttunen, C. D., and Stuckman, B. E. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications 79*, 1 (1993), 157–181.

[108] Jones, D. R., Schonlau, M., and Welch, W. J. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization 13* (1998), 455–492.

[109] Kaleva, O. Fuzzy differential equations. *Fuzzy Sets and Systems 24* (1987), 301–317.

[110] Kaleva, O. The Cauchy problem for fuzzy differential equations. *Fuzzy Sets and Systems 35*, 3 (1990), 389–396.

[111] Kaufmann, A., and Gupta, M. M. *Introduction to fuzzy arithmetic: theory and applications.* Van Nostrand Reinhold, New York, 1985.

[112] Keynes, J. M. *A treatise on probability.* Macmillan & Co., London, 1921.

[113] Khoromskij, B. N. Tensors-structured numerical methods in scientific computing: survey on recent advances. *Chemometrics and Intelligent Laboratory Systems 110*, 1 (2012), 1–19.

[114] Khoromskij, B. N., and Oseledets, I. Quantics-TT collocation approximation of parameter-dependent and stochastic elliptic PDEs. *Computational Methods in Applied Mathematics 10*, 4 (2010), 376–394.

[115] Khoromskij, B. N., and Schwab, C. Tensor-structured Galerkin approximation of parametric and stochastic elliptic PDEs. *SIAM Journal on Scientific Computing 33*, 1 (2011), 364–385.

[116] Kilmer, M. E., and de Sturler, E. Recycling subspace information for diffuse optical tomography. *SIAM Journal on Scientific Computing 27*, 6 (2006), 2140–2166.

[117] Kirsch, U. A unified reanalysis approach for structural analysis, design, and optimization. *Structural and Multidisciplinary Optimization 25*, 2 (2003), 67–85.

[118] Kirsch, U., Kocvara, M., and Zowe, J. Accurate reanalysis of structures by a preconditioned conjugate gradient method. *International Journal for Numerical Methods in Engineering 55*, 2 (2002), 233–251.

[119] Klimke, A., Nunes, R. F., and Wohlmuth, B. I. Fuzzy arithmetic based on dimension-adaptive sparse grids: a case study of a large-scale finite element model under uncertain parameters. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 14*, 5 (2006), 561–577.

[120] Klimke, A., and Wohlmuth, B. Computing expensive multivariate functions of fuzzy numbers using sparse grids. *Fuzzy Sets and Systems 153*, 3 (2005), 432–453.

[121] Klimke, W. A. *Uncertainty modeling using fuzzy arithmetic and sparse grids*. PhD thesis, Universität Stuttgart, Stuttgart, Germany, 2006.

[122] Klir, G. J. Generalized information theory: aims, results, and open problems. *Reliability Engineering & System Safety 85*, 1-3 (2004), 21–38.

[123] Klir, G. J. *Uncertainty and information: foundations of generalized information theory*. John Wiley & Sons Inc., Hoboken, NJ, 2006.

[124] Kolda, T. G., and Bader, B. W. Tensor decompositions and applications. *SIAM Review 51*, 3 (2009), 455–500.

[125] Kressner, D., and Tobler, C. Low-rank tensor Krylov subspace methods for parametrized linear systems. *SIAM Journal on Matrix Analysis and Applications 32*, 4 (2011), 1288–1316.

[126] Kressner, D., and Tobler, C. Preconditioned low-rank methods for high-dimensional elliptic PDE eigenvalue problems. *Computational Methods in Applied Mathematics 11*, 3 (2011), 363–381.

[127] Kressner, D., and Tobler, C. htucker - A Matlab toolbox for tensors in hierarchical Tucker format, 2012.

[128] Kroshko, D. OpenOpt: free scientific-engineering software for mathematical modeling and optimization, 2013.

[129] Laksmikantham, V. Set differential equations versus fuzzy differential equations. *Applied Mathematics and Computation 164*, 2 (2005), 277–294.

[130] Lasserre, J. B. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization 11*, 3 (2001), 796–817.

[131] Lasserre, J. B. Moments and sums of squares for polynomial optimization and related problems. *Journal of Global Optimization 45*, 1 (2009), 39–61.

[132] Lasserre, J. B. *Moments, positive polynomials and their applications*. Imperial College Press, London, 2010.

[133] Le Maître, O. P., Knio, O. M., Debusschere, B. J., Najm, H. N., and Ghanem, R. G. A multigrid solver for two-dimensional stochastic diffusion equations. *Computer Methods in Applied Mechanics and Engineering 192*, 41-42 (2003), 4723–4744.

[134] Le Maître, O. P., Knio, O. M., Najm, H. N., and Ghanem, R. G. Uncertainty propagation using Wiener–Haar expansions. *Journal of Computational Physics 197*, 1 (2004), 28–57.

[135] Le Maître, O. P., Najm, H. N., Ghanem, R. G., and Knio, O. M. Multi-resolution analysis of Wiener-type uncertainty propagation schemes. *Journal of Computational Physics 197*, 2 (2004), 502–531.

[136] Lee, D. H., and Park, D. An efficient algorithm for fuzzy weighted average. *Fuzzy Sets and Systems 87*, 1 (1997), 39–45.

[137] Leland, R. P. Fuzzy differential systems and Malliavin calculus. *Fuzzy Sets and Systems 70* (1995), 59–73.

[138] Levi, I. *The enterprise of knowledge: an essay on knowledge, credal probability, and chance*. The MIT Press, Cambridge, MA, 1983.

[139] Liang, J. J., Suganthan, P. N., and Deb, K. Novel composition test functions for numerical global optimization. In *Proceedings of Swarm Intelligence Symposium, 2005* (Pasadena, CA, 2005), IEEE, pp. 68–75.

[140] Lodwick, W. A., and Jamison, K. D. Interval-valued probability in the analysis of problems containing a mixture of possibilistic, probabilistic, and interval uncertainty. *Fuzzy Sets and Systems 159*, 21 (2008), 2845–2858.

[141] Ma, X., and Zabaras, N. An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations. *Journal of Computational Physics 228*, 8 (2009), 3084–3113.

[142] Marshall, M. Representations of non-negative polynomials, degree bounds and applications to optimization. *Canadian Journal of Mathematics 61* (2009), 205–221.

[143] Mason, J. C., and Handscomb, D. C. *Chebyshev polynomials*. CRC Press Company, Boca Raton, FL, 2003.

[144] Matthies, H. G., and Keese, A. Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations. *Computer Methods in Applied Mechanics and Engineering 194*, 12-16 (2005), 1295–1331.

[145] Matthies, H. G., and Zander, E. Solving stochastic systems with low-rank tensor compression. *Linear Algebra and its Applications 436*, 10 (2012), 3819–3838.

[146] Maute, K., Weickum, G., and Eldred, M. A reduced-order stochastic finite element approach for design optimization under uncertainty. *Structural Safety 31*, 6 (2009), 450–459.

[147] Mizukoshi, M. T., Barros, L. C., Chalco-Cano, Y., Román-Flores, H., and Bassanezi, R. C. Fuzzy differential equations and the extension principle. *Information Sciences 177*, 17 (2007), 3627–3635.

[148] Moens, D., De Munck, M., Desmet, W., and Vandepitte, D. Numerical dynamic analysis of uncertain mechanical structures based on interval fields. In *IUTAM Symposium on the Vibration Analysis of Structures with Uncertainties* (Dordrecht, The Netherlands, 2011), A. K. Belyaev and R. S. Langley, Eds., vol. 27 of *IUTAM Bookseries*, Springer Netherlands, pp. 71–83.

[149] Moens, D., and Hanss, M. Non-probabilistic finite element analysis for parametric uncertainty treatment in applied mechanics: Recent advances. *Finite Elements in Analysis and Design 47*, 1 (2011), 4–16.

[150] Moens, D., and Vandepitte, D. Fuzzy finite element method for frequency response function analysis of uncertain structures. *AIAA journal 40*, 1 (2002), 126–136.

[151] Moens, D., and Vandepitte, D. A fuzzy finite element procedure for the calculation of uncertain frequency-response functions of damped structures: Part 1—Procedure. *Journal of Sound and Vibration 288*, 3 (2005), 431–462.

[152] Moens, D., and Vandepitte, D. A survey of non-probabilistic uncertainty treatment in finite element analysis. *Computer Methods in Applied Mechanics and Engineering 194*, 12-16 (2005), 1527–1555.

[153] Möller, B., and Beer, M. *Fuzzy randomness: uncertainty in civil engineering and computational mechanics*. Springer–Verlag, Heidelberg, Germany, 2004.

[154] Möller, B., and Beer, M. Engineering computation under uncertainty—Capabilities of non-traditional models. *Computers & Structures 86*, 10 (2008), 1024–1041.

[155] Moore, R. E. *Interval analysis*. Prentice-Hall, Englewood Cliffs, NJ, 1966.

[156] Moré, J. J., and Wild, S. M. Benchmarking derivative-free optimization algorithms. *SIAM Journal on Optimization 20*, 1 (2009), 172–191.

[157] Narayan, A., and Xiu, D. Stochastic collocation methods on unstructured grids in high dimensions via interpolation. *SIAM Journal on Scientific Computing 34*, 3 (2012), 1729–1752.

[158] Neumaier, A. Complete search in continuous global optimization and constraint satisfaction. *Acta Numerica 13* (2004), 1–94.

[159] Nguyen, H. T. A note on the extension principle for fuzzy sets. *Journal of Mathematical Analysis and Applications 64* (1978), 369–380.

[160] Nicolaï, B. M., Egea, J. A., Scheerlinck, N., Banga, J. R., and Datta, A. K. Fuzzy finite element analysis of heat conduction problems with uncertain parameters. *Journal of Food Engineering 103*, 1 (2011), 38–46.

[161] Nie, J., and Schweighofer, M. On the complexity of Putinar's Positivstellensatz. *Journal of Complexity 23*, 1 (2007), 135–150.

[162] Nobile, F., Tempone, R., and Webster, C. G. A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM Journal on Numerical Analysis 46*, 5 (2008), 2309–2345.

[163] Nobile, F., Tempone, R., and Webster, C. G. An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM Journal on Numerical Analysis 46*, 5 (2008), 2411–2442.

[164] Novak, E. *Deterministic and stochastic error bounds in numerical analysis*. Springer Verlag, Berlin, 1988.

[165] Oberguggenberger, M., and Pittschmann, S. Differential equations with fuzzy parameters. *Mathematical and Computer Modelling of Dynamical Systems 5*, 3 (1999), 181–202.

[166] Oberkampf, W. L., DeLand, S. M., Rutherford, B. M., Diegert, K. V., and Alvin, K. F. Error and uncertainty in modeling and simulation. *Reliability Engineering & System Safety 75*, 3 (2002), 333–357.

[167] Oseledets, I. V., and Dolgov, S. V. Solution of linear systems and matrix inversion in the TT-format. *SIAM Journal on Scientific Computing 34*, 5 (2012), 2718–2739.

[168] Parks, M. L., de Sturler, E., Mackey, G., Johnson, D. D., and Maiti, S. Recycling Krylov Subspaces for Sequences of Linear Systems. *SIAM Journal on Scientific Computing 28*, 5 (2006), 1651–1674.

[169] Parrilo, P. A. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. PhD thesis, California Institute of Technology, Pasadena, CA, 2000.

[170] Parrilo, P. A. Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming 96*, 2 (2003), 293–320.

[171] Parrilo, P. A., and Sturmfels, B. Minimizing polynomial functions. *Algorithmic and Quantitative Real Algebraic Geometry 60* (2003), 83–99.

[172] Peirce, C. S. *Collected papers of Charles Sanders Peirce, vols. 7–8*. Harvard University Press, Cambridge, MA, 1958.

[173] Popper, K. R. *A world of propensities*. Thoemmes Press, Bristol, UK, 1990.

[174] Powell, C. E., and Elman, H. C. Block-diagonal preconditioning for spectral stochastic finite-element systems. *IMA Journal of Numerical Analysis 29*, 2 (2009), 350–375.

[175] Puri, M. L., and Ralescu, D. A. Differentials of fuzzy functions. *Journal of Mathematical Analysis and Applications 91*, 2 (1983), 552–558.

[176] Ramsey, F. P. Truth and probability. In *The foundations of mathematics and other logical essays*, R. B. Braithwaite, Ed. Harcourt, Brace and Company, New York, 1931, ch. VII, pp. 156–198.

[177] Rao, S., and Sawyer, J. P. Fuzzy finite element approach for analysis of imprecisely defined systems. *AIAA journal 12*, 33 (1995), 2364–2370.

[178] Regis, R. G., and Shoemaker, C. A. A stochastic radial basis function method for the global optimization of expensive functions. *INFORMS Journal on Computing 19*, 4 (2007), 497–509.

[179] Rios, L. M., and Sahinidis, N. V. Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization 56* (2013), 1247–1293.

[180] Ross, T. J. *Fuzzy logic with engineering applications*. John Wiley & Sons, Ltd., Chichester, UK, 2010.

[181] Rosseel, E., Boonen, T., and Vandewalle, S. Algebraic multigrid for stationary and time-dependent partial differential equations with stochastic coefficients. *Numerical Linear Algebra With Applications 15*, 2008 (2008), 141–163.

[182] Rosseel, E., and Vandewalle, S. Iterative solvers for the stochastic finite element method. *SIAM Journal on Scientific Computing 32*, 1 (2010), 372–397.

[183] Scheerlinck, K., Vernieuwe, H., and De Baets, B. Zadeh's extension principle for continuous functions of non-interactive variables: a parallel optimization approach. *IEEE Transactions on Fuzzy Systems 20*, 1 (2012), 96–108.

[184] Schweighofer, M. On the complexity of Schmüdgen's Positivstellensatz. *Journal of Complexity 20*, 4 (2004), 529–543.

[185] Seikkala, S. On the fuzzy initial value problem. *Fuzzy Sets and Systems 24* (1987), 319–330.

[186] Seynaeve, B., Rosseel, E., Nicolaï, B., and Vandewalle, S. Fourier mode analysis of multigrid methods for partial differential equations with random coefficients. *Journal of Computational Physics 224*, 1 (2007), 132–149.

[187] Shafer, G. *A mathematical theory of evidence*, vol. 1. Princeton university press, Princeton, NJ, 1976.

[188] Shan, S., and Wang, G. G. Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. *Structural and Multidisciplinary Optimization 41*, 2 (2010), 219–241.

[189] Simoncini, V., and Szyld, D. B. Recent computational developments in Krylov subspace methods for linear systems. *Numerical Linear Algebra with Applications 14*, 1 (2007), 1–59.

[190] Smolyak, S. A. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Soviet Mathematics - Doklady 4* (1963), 240–243.

[191] Spall, J. C. *Introduction to stochastic search and optimization: estimation, simulation, and control*. John Wiley & Sons Inc., Hoboken, NJ, 2003.

[192] Stüben, K. A review of algebraic multigrid. *Journal of Computational and Applied Mathematics 128* (2001), 281–309.

[193] Sun, W., and Yuan, Y.-X. *Optimization theory and methods: nonlinear programming*. Springer Science+Business Media, LLC, New York, 2006.

[194] Todor, R. A., and Schwab, C. Convergence rates for sparse chaos approximations of elliptic problems with stochastic coefficients. *IMA Journal of Numerical Analysis 27*, 2 (2006), 232–261.

[195] Tomlab Optimization Inc. User's guide for TOMLAB, 2013.

[196] Trefethen, L. N. *Approximation theory and approximation practice*. SIAM, Philadelphia, PA, 2013.

[197] Trottenberg, U., Oosterlee, C., and Schüller, A. *Multigrid*. Academic Press, London, 2001.

[198] Tucker, L. R. Some mathematical notes on three-mode factor analysis. *Psychometrika 31*, 3 (1966), 279–311.

[199] Ullmann, E., Elman, H. C., and Ernst, O. G. Efficient iterative solvers for stochastic Galerkin discretizations of log-transformed random diffusion problems. *SIAM Journal on Scientific Computing 34*, 2 (2012), 659–682.

[200] Vaněk, P., Mandel, J., and Brezina, M. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. *Computing 53*, 3 (1996), 176–196.

[201] Vavasis, S. A. *Nonlinear optimization: complexity issues*. Oxford University Press, Inc., New York, 1991.

[202] Venn, J. *The logic of chance*. Macmillan & Co., London, 1888.

[203] Verhaeghe, W., Desmet, W., Vandepitte, D., and Moens, D. Interval fields to represent uncertainty on the output side of a static FE analysis. *Computer Methods in Applied Mechanics and Engineering 260* (2013), 50–62.

[204] Vorobiev, D., and Seikkala, S. Towards the theory of fuzzy differential equations. *Fuzzy Sets and Systems 125* (2002), 231–237.

[205] Walley, P. *Coherent lower (and upper) probabilities*. Statistics Research Report 22, University of Warwick, Conventry, UK, 1981.

[206] Walley, P. *Statistical reasoning with imprecise probabilities*. Chapman and Hall, London, 1991.

[207] Walley, P. Measures of uncertainty in expert systems. *Artificial intelligence 83* (1996), 1–58.

[208] Walley, P. Towards a unified theory of imprecise probability. *International Journal of Approximate Reasoning 24* (2000), 125–148.

[209] Wan, X., and Karniadakis, G. E. An adaptive multi-element generalized polynomial chaos method for stochastic differential equations. *Journal of Computational Physics 209*, 2 (2005), 617–642.

[210] Wan, X., and Karniadakis, G. E. Solving elliptic problems with non-Gaussian spatially-dependent random coefficients. *Computer Methods in Applied Mechanics and Engineering 198*, 21-26 (2009), 1985–1995.

[211] Wang, S., de Sturler, E., and Paulino, G. H. Large-scale topology optimization using preconditioned Krylov subspace methods with recycling. *International Journal for Numerical Methods in Engineering 69* (2007), 2441–2468.

[212] Weichselberger, K. The theory of interval-probability as a unifying concept for uncertainty. *International Journal of Approximate Reasoning 24*, 2-3 (2000), 149–170.

[213] Wienands, R., and Joppich, W. *Practical Fourier analysis for multigrid methods*. CRC Press, Boca Raton, FL, 2005.

[214] Xiu, D. Fast numerical methods for stochastic computations: a review. *Communications in Computational Physics 5*, 2 (2009), 242–272.

[215] Xiu, D., and Hesthaven, J. S. High-order collocation methods for differential equations with random inputs. *SIAM Journal on Scientific Computing 27*, 3 (2005), 1118–1139.

[216] Xiu, D., and Karniadakis, G. E. The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing 24*, 2 (2002), 619–644.

[217] Yang, H. Q., Yao, H., and Jones, J. D. Calculating functions of fuzzy numbers. *Fuzzy Sets and Systems 55*, 3 (1993), 273–283.

[218] Zadeh, L. A. Fuzzy sets. *Information and Control 8*, 3 (1965), 338–353.

[219] Zadeh, L. A. The concept of a linguistic variable and its application to approximate reasoning—I. *Information Sciences 8*, 3 (1975), 199–249.

[220] Zadeh, L. A. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems 1* (1978), 3–28.

[221] Zadeh, L. A. Generalized theory of uncertainty (GTU)—principal concepts and ideas. *Computational Statistics & Data Analysis 51*, 1 (2006), 15–46.

Curriculum vitae

Samuel Corveleyn
° August 7, 1979, Leuven (Belgium)

Higher education

2008-2014 Ph.D. in Engineering, Katholieke Universiteit Leuven, Belgium
Thesis: *The numerical solution of elliptic partial differential equations with fuzzy coefficients*

2003-2005 Candidate Composing Music: Music Production, Hogeschool Gent, Belgium

1997-2002 Civil Electrotechnical Engineer, Katholieke Universiteit Leuven, Belgium
Thesis: *Speech modification: conversion of pitch and articulation*

Work history

02/2008-03/2008 Audio Engineer, Philips Consumer Electronics, Oud-Heverlee, Belgium

10/2005-12/2007 Research assistant, Department of Electrical Engineering, Katholieke Universiteit Leuven, Belgium

10/2002-03/2003 Research assistant, Department of Electronics and Informatics, Vrije Universiteit Brussel, Belgium

Publications in reviewed journals

- Corveleyn, S., Rosseel, E., And Vandewalle, S. Iterative solvers for a spectral Galerkin approach to elliptic partial differential equations with fuzzy coefficients. *SIAM Journal on Scientific Computing 35*, 5 (2013), S420–S444.
- Corveleyn, S., And Vandewalle, S. Application of a low-rank tensor approximation for the fuzzification of a function. *Fuzzy Sets and Systems*, (2013, submitted).

Publications in conference proceedings

- Corveleyn, S., And Vandewalle, S. A polynomial response surface approach for the solution of fuzzy elliptic partial differential equations. In *Proceedings of the 8th International Conference on Structural Dynamics* (Leuven, Belgium, 2011), pp. 3048–3055.
- Corveleyn, S., And Vandewalle, S. On the numerical solution of fuzzy elliptic PDEs by means of polynomial response surfaces. In *Proceedings of the 24th International Conference on Noise and Vibration Engineering* (Leuven, Belgium, 2010), pp. 5007–5014.
- Corveleyn, S., And Vandewalle, S. Component reuse in iterative solvers for the solution of fuzzy partial differential equations. In *Proceedings of the 7th International Conference of Numerical Analysis and Applied Mathematics* (Rethymno, Greece, 2009), pp. 452–455.
- Corveleyn, S., Coose, B., Verhelst, W. Voice modification and conversion using PLAR-parameters. In *Proceedings of the 1st IEEE Benelux Workshop on Model Based Processing and Coding of Audio (MPCA-2002)* (Leuven, Belgium, 2002), pp. 37–40.

Conference presentations

- Corveleyn, S., And Vandewalle, S. *A polynomial response surface approach for the solution of fuzzy elliptic partial differential equations.* Eighth International Conference on Structural Dynamics, Leuven, Belgium, July 4–6, 2011.

- Corveleyn, S., And Vandewalle, S. *Solving fuzzy elliptic PDEs by means of HT tensors.* Fourth Workshop on High-Dimensional Approximation, Bonn, Germany, June 26–30, 2011.
- Corveleyn, S., And Vandewalle, S. *Solving fuzzy PDEs by a polynomial chaos expansion.* Eighty-second GAMM Annual Meeting, Graz, Austria, April 18–21, 2011.
- Corveleyn, S., And Vandewalle, S. *On the numerical solution of fuzzy elliptic PDEs by means of polynomial response surfaces.* SIAM Conference on Computational Science and Engineering, Reno, Nevada, February 28–March 4, 2011.
- Corveleyn, S., And Vandewalle, S. *On the numerical solution of fuzzy elliptic PDEs by means of polynomial response surfaces.* Third International Conference on Uncertainty in Structural Dynamics, Leuven, Belgium, September 20–22, 2010.
- Corveleyn, S., And Vandewalle, S. *Solution of fuzzy elliptic PDEs by a polynomial chaos expansion.* Fifteenth International Congress on Computational and Applied Mathematics, Leuven, Belgium, July 5–9, 2010.
- Corveleyn, S., And Vandewalle, S. *Component reuse in iterative solvers for the solution of fuzzy partial differential equations.* Seventh International Conference of Numerical Analysis and Applied Mathematics, Rethymno, Greece, September 18–22, 2009.

Seminars

- *A new approach to solve fuzzy PDEs using polynomial chaos.* Doctoral Seminar, Katholieke Universiteit Leuven, Belgium, January 29, 2010.

Teaching

2008-2009 Teaching assistant for *Hogere Wiskunde*

2008-2013 Teaching assistant for *Probleemoplossen en Ontwerpen, deel 1*

2008-2010 Teaching assistant for *Technisch-wetenschappelijke rekenomgevingen*

2009-2011 Teaching assistant for *Lineaire algebra*

2010-2013 Teaching assistant for *Analyse, deel 3*

2012-2013 Teaching assistant for *Probleemoplossen en Ontwerpen, ICT-werktuigen*

FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF COMPUTER SCIENCE
NUMERICAL ANALYSIS AND APPLIED MATHEMATICS DIVISION
Celestijnenlaan 200A box 2402
B-3001 HEVERLEE, BELGIUM